Behavioural Theory for Mobile Ambients

Massimo Merro, Francesco Zappa Nardelli

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Behavioural Theory for Mobile Ambients

Massimo Merro
Dipartimento di Informatica
Università di Verona, Italy
Massimo.Merro@univr.it

Francesco Zappa Nardelli
INRIA Rocquencourt
France
Francesco.Zappa_Nardelli@inria.fr

No 5375

November 2004

Thème COM

ISSN 0249-6399

INRIA
ROCQUENCOURT

Behavioural Theory for Mobile Ambients

Massimo Merro
Dipartimento di Informatica
Università di Verona, Italy
Massimo.Merro@univr.it

Francesco Zappa Nardelli
INRIA Rocquencourt
France
Francesco.Zappa_Nardelli@inria.fr

Thème COM — Systèmes communicants
Projet Moscova

Rapport de recherche no 5375 — November 2004 — 58 pages

Abstract: We study a behavioural theory of Cardelli and Gordon's *Mobile Ambients*, a process calculus for modelling mobile agents in wide-area networks, focussing on *reduction barbed congruence*. Our contribution is threefold. (1) We prove a *context lemma* that shows that only parallel and nesting contexts need be examined to recover this congruence. (2) We characterise this congruence using a *labelled bisimilarity*: this requires novel techniques to deal with asynchronous movements of agents and with the invisibility of migrations of secret locations. (3) We develop refined proof methods involving *up-to proof techniques*, which allow us to verify a set of *algebraic laws* and the correctness of more complex examples.

Key-words: Programming languages, concurrency, process calculi, behavioural theories, bisimulation

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Une théorie comportementale pour les Ambients Mobiles

Résumé : Nous étudions une théorie comportementale des Ambients Mobiles, calcul des processus proposé par Cardelli et Gordon pour modéliser des agents mobiles. Notre contribution prend pour point de départ la *congruence barbue fermée par réduction*, équivalence contextuelle naturelle, et se structure en trois volée. (1) Nous démontrons un *lemme de contexte*, selon lequel la congruence barbue fermée par réduction est influencée par les seuls contextes de composition parallèle des agents et d'imbrication des localisations. (2) Nous donnons une bisimulation étiquetée qui caractérise complètement la congruence barbue fermée par réduction. Ce résultat nécessite l'élaboration de techniques spécifiques permettant de manipuler la mobilité asynchrone et le *stuttering*. (3) Nous enrichissons nos techniques par des méthodes de preuve dites *up-to*. Nos résultats permettent de vérifier aisément des *lois algébriques* du calcul des Ambients Mobiles, ainsi que de montrer la correction d'applications plus complexes.

Mots-clés : Langages de programmation, concurrence, calculs des processus, théories comportementales; bisimulation

Introduction

The calculus of *Mobile Ambients*, abbreviated MA, has been introduced by Cardelli and Gordon, [6], as a process calculus for describing *mobile agents*.

In MA, the term $n[P]$ represents an agent, or *ambient*, named n, executing the code P; the ambient n is a bounded, protected, and (potentially) mobile space where the computation P takes place. In turn P may contain other ambients, may perform (local) *communications*, or may exercise *capabilities*, that allow entry to or exit from named ambients, and to dissolve ambient's boundaries. *Ambient names*, such as n, are used to control access to the ambient's computation space and may be dynamically created as in the π-calculus, [27], using the construct $(\boldsymbol{\nu} n)P$.

A central concern in the theory of concurrent process calculi is to establish when two processes have the same *observable behaviour*, independently on the environment they are located. *Behavioural equalities* are used to justify program transformations performed either by programmers, during system development, or by the optimising phases of compilers. Several notions of behavioural equalities can be found in the literature; among these, *testing equivalence*, [8], and *bisimulation equivalence*, [29], have emerged as widely accepted equivalences to define semantic theories for a variety of process calculi such as CCS, [25], and the π-calculus, [27].

In this paper we focus on bisimulation-based equivalences. Our touchstone behavioural equality is *reduction barbed congruence*, a slight variant of Milner and Sangiorgi's *barbed congruence* [28] also called *open barbed bisimilarity* [37]. Reduction barbed congruence was first studied by Honda and Yoshida for the π-calculus under the name of *maximum sound theory* [18]. Reduction barbed congruence is a context-based equivalence widely-used in concurrent process calculi for its simple and intuitive definition. More precisely, it is the largest equivalence relation that

- is preserved by the constructs of the language;
- preserves, in some sense, the *reduction semantics* of the language, i.e., the evolution of processes;
- preserves *barbs*, simple observational properties of terms.

Context-based behavioural equalities, such as reduction barbed congruence, involve a universal quantification on all contexts; thus direct proofs of process equalities are often very hard. Simpler proof techniques are based on *labelled bisimilarities*, co-inductive relations that characterise the behaviour of processes using a *labelled transition system*, or *LTS*, a collection of relations of the form

$$P \xrightarrow{\alpha} Q.$$

Intuitively, the action α in the judgement $P \xrightarrow{\alpha} Q$ represents some small context process P can interact with; if the labelled bisimilarity coincides with reduction barbed congruence [30, 1, 10] then this collection of small contexts, codified as actions, captures all the interactions that processes can have with arbitrary contexts.

Although the idea of bisimulation is very general and does not rely on the specific syntax of the calculus, the definition of an appropriate notion of bisimilarity for Mobile Ambients revealed harder than expected. The reasons of that can be resumed as follows:

- In general, an ambient n suffers from *interferences* that may originate either from other ambients of its environment or from the computation running at n itself, [21].

- *Ambient mobility is asynchronous* — no synchronisation is required to migrate into an ambient. As noticed by Sangiorgi, [34], this causes a *stuttering* phenomenon originated by ambients that may repeatedly enter and exit another ambient. Stuttering cannot be observed by reduction barbed congruence, and any successful labelled characterisation of reduction barbed congruence should not observe stuttering as well [34]. As an example, the two processes:[1]

$$P \stackrel{\text{def}}{=} \texttt{in}\_n.\texttt{out}\_n.\texttt{in}\_n.R \text{ and } Q \stackrel{\text{def}}{=} \texttt{in}\_n.\texttt{out}\_n.\texttt{in}\_n.R + \texttt{in}\_n.R$$

  cannot be distinguished by reduction barbed congruence. Process Q can obviously simulate process P; however, also P can simulate Q. For instance, P can mimic the reduction $k[Q] \mid n[\ ] \rightarrowtriangle n[k[R]]$ by performing three consecutive reductions: $k[P] \mid n[\ ] \rightarrowtriangle \rightarrowtriangle \rightarrowtriangle n[k[R]]$.

- Finally, consider the *perfect firewall equation*, [6], a well-known algebraic law of MA:

$$(\boldsymbol{\nu} n)n[P] = \mathbf{0} \qquad \text{for } n \text{ not in } P.$$

  This law states that a *private ambient* n whose internal code does not refer to the name of the ambient itself, is equivalent to the inactive process. The subtle point is that the ambient n can freely move around the network without being observed. As a consequence, a bisimilarity that wants to capture this law must not observe the movements of private ambients.

Merro and Hennessy, [23], introduced a labelled bisimilarity for a simpler variant of MA, called SAP, equipped with (i) *synchronous mobility*, as in Levi and Sangiorgi's *Safe Ambients* [21], and (ii) *passwords* to exercise control over, and differentiate between, different ambients that wish to exercise a capability. Their main result is the characterisation of reduction barbed congruence in terms of the labelled bisimilarity. The result holds only in SAP and crucially relies on the two features (i) and (ii) mentioned above.

This paper is the natural continuation of Merro and Hennessy investigations, where we tackle the original problem: *to provide bisimulation proof methods for Mobile Ambients.*

Road Map The aim of this work is to provide a labelled characterisation of *reduction barbed congruence over processes*. This is achieved by a careful study of the behavioural theory for a wide class of processes, called *systems*. We outline the main contributions of this paper, highlighting how they fit together.

Section 1 First of all, as in the Distributed π-calculus, [15], we divide MA terms in two categories: *processes* and *systems*. Processes denote threads, whereas systems consist of collections of ambients, running in parallel, that may share the knowledge of ambient names; technically speaking, systems are processes that do not exercise capabilities at top-level.

Section 2 We define a labelled transition system for systems.

[1] For simplicity we use guarded choice *à la* CCS; the same phenomenon can be exhibited using replication.

Section 3 On top of the LTS, we define a labelled bisimilarity over systems. Technically speaking, this is a *context bisimulation*, [31], as it involves a universal quantification over processes provided by the environment. However, contrarily to Sangiorgi's context bisimulation, the contexts we use in the co-inductive step are very simple. Depending on the position of this quantification in the definition of bisimulation, we can define both *late* and *early* bisimilarity. As in HOπ, [31], the two formulations coincide, and we concentrate on the late version, $\approx$, which is easier to study. The definition of $\approx$ reminds us the asynchronous bisimilarity of Amadio, Castellani and Sangiorgi for Asynchronous π-calculus, [1]. More precisely, our bisimilarity does not observe the movements of secret ambients, in the same way as asynchronous bisimilarity does not observe input actions.

We prove that the relation $\approx$ completely characterises *reduction barbed congruence over systems*, $\cong_{\rm s}$, that is, for all systems M and N it holds that

$$M \approx N \text{ iff } M \cong_{\rm s} N \ .$$

Section 4 We provide two *up-to proof techniques*, along the lines of [28, 33, 36]. More precisely, we develop both *up-to expansion* and *up-to context* proof techniques for $\approx$, and prove their soundness. As $\approx$ is a context bisimulation, the up-to-context proof-technique reveals to be very useful to factor out the processes provided by the environment. We are not aware of other forms of up-to proof techniques for higher-order calculi.

Section 5 We then use the theory developed for systems to characterise reduction barbed congruence over processes, $\cong_{\rm p}$, in terms of $\approx$. More precisely, we show that:

$$\cong_{\rm p} = \ \{(P,Q) : \ k[P \mid R] \approx k[Q \mid R] \text{ for all } k,\ R\}$$

where P and Q range over processes. This result relies crucially on a *context lemma* for $\cong_{\rm p}$, which allows us to consider only contexts for concurrency and locality.

When restricting our attention to systems, a stronger results holds: for all systems M and N it holds that

$$M \approx N \text{ iff } M \cong_{\rm p} N \ .$$

This result, together with that in Section 3, shows that system contexts have the same distinguishing power of the more general processes contexts.

Section 6 We extend our results to the full calculus of Mobile Ambients processes equipped with communication of capabilities. A consequence of constructing our proof methods on top of the behaviour of systems is that very little modifications are required to accommodate communication.

Section 7 We put our bisimulation proof methods at work proving a collection of *algebraic laws* (among which the *perfect firewall equation* [6]) with respect to $\cong_{\rm p}$. The proofs are pleasantly simple: the size of the required bisimulations is small thanks to the up-to context proof technique. We also prove the correctness of a protocol, introduced in [6], for controlling access through a firewall.

The paper ends with a comparison with related work.

Table 1 Mobile Ambients in Two Levels

Names: $a, b, \ldots, k, l, m, n, \ldots \in \mathbf{N}$

Systems:

M, N	::=	$\mathbf{0}$	inactive system
	$\mid$	$M_1 \mid M_2$	parallel composition
	$\mid$	$(\boldsymbol{\nu} n)M$	restriction
	$\mid$	$n[P]$	ambient

Processes:

P, Q, R	::=	$\mathbf{0}$	inactive process
	$\mid$	$P_1 \mid P_2$	parallel composition
	$\mid$	$(\boldsymbol{\nu} n)P$	restriction
	$\mid$	$C.P$	prefixing
	$\mid$	$n[P]$	ambient
	$\mid$	$!C.P$	replication

Capabilities:

C	::=	$\texttt{in\_}n$	may enter into n
	$\mid$	$\texttt{out\_}n$	may exit out of n
	$\mid$	$\texttt{open\_}n$	may open n

1 Mobile Ambients in Two Levels

In Table 1 we report the syntax of MA, where $\mathbf{N}$ denotes a countable infinite set of names.

Unlike the original definitions of MA, our syntax is defined in a two-level structure, a lower one for *processes*, and an upper one for *systems*. Systems are collections of ambients running in parallel, that may share knowledge of ambient names. As regards processes, the constructs for inactivity, parallel composition, restriction and replicated prefixing are inherited from mainstream concurrent calculi, most notably the π-calculus [27]. The inactive process, $\mathbf{0}$, does nothing. Parallel composition is denoted by the commutative and associative operator, $P \mid Q$. The restriction operator, $(\boldsymbol{\nu} n)P$, creates a new fresh name n within a scope P. We have replicated prefixing, $!C.P$, (rather than full replication) to create as many parallel replicas of a process as needed. As in the π-calculus replicated prefixing allows us to derive a simpler LTS, and to work with simpler proofs. We also recall that in the π-calculus (i) replicated input has the same expressive power as full replication [17] and recursion [26, 36]; (ii) replicated input has a simpler semantics and is handy for implementations.

Specific of the ambient calculus are the *ambient* construct, $n[P]$, and the *prefixing* of capabilities, $C.P$. In $n[P]$, n is the name of the ambient and P is the process running inside the ambient. The process $C.P$ performs an action regulated by the capability C, and then continues as the process P. Capabilities are constructed from names; given a name n, the capability $\texttt{in\_}n$ allows entry into n, the capability $\texttt{out\_}n$ allows exit out of n, and the capability $\texttt{open\_}n$ allows the destruction of the boundary of ambient n. To avoid unnecessary complications at this stage, we omit *communication*; it will be added in Section 6.

A (monadic) *context* $C[-]$ is a process with a hole inside denoted by $-$. A *static context* is a context where the hole does not appear under a prefix or a replication. The class of systems is not closed under arbitrary contexts, so we call *systems contexts* those static contexts that transform systems into systems. Formally, system contexts are generated by the following grammar:

$$C[-] \;::=\; - \;\mid\; C[-] \mid M \;\mid\; M \mid C[-] \;\mid\; (\boldsymbol{\nu} n)C[-] \;\mid\; n[C[-] \mid P] \;\mid\; n[P \mid C[-]]$$

where M is an arbitrary system, and P is an arbitrary process. System contexts, as we will see in Section 5, have the same distinguishing power as arbitrary contexts. The contexts exhibited in the paper will be always monadic, unless otherwise specified.

We use a number of notational conventions. Parallel composition has the lowest precedence among the operators. $\prod_{i\in I} P_i$ means the parallel composition of all processes P_i, for $i \in I$. $\tilde{n}$ denotes a tuple $n_1, \ldots, n_k$ of names. The process $C.C'.P$ is read as $C.(C'.P)$. We omit trailing dead processes, writing C for $C.\mathbf{0}$, and $n[\,]$ for $n[\mathbf{0}]$. The operator $(\boldsymbol{\nu} n)$ is a binder for names, leading to the usual notions of free and bound occurrences of names, $\mathrm{fn}(\cdot)$ and $\mathrm{bn}(\cdot)$, and α-conversion, $\equiv_\alpha$. We write $(\boldsymbol{\nu}\tilde{n})P$ as an abbreviation for $(\boldsymbol{\nu} n_1)\ldots(\boldsymbol{\nu} n_k)P$. We will identify processes up to α-conversion. More formally we will view process terms as representatives of their equivalence class with respect to $\equiv_\alpha$, and these representatives will always be chosen so that bound names are distinct from free names.

Operational semantics The dynamics of the calculus is specified by the *reduction relation* over processes, $\rightarrowtriangle$, described in Table 2. As systems are processes with a special structure, the rules of Table 2 also describe the evolution of systems. The *reduction semantics* relies on an auxiliary relation called *structural congruence* that brings the participants of a potential interaction into contiguous positions. It is easy to check that the class of systems is closed under the reduction relation, that is, systems always reduce to systems. The symbol $\rightarrowtriangle^*$ denotes the reflexive and transitive closure of $\rightarrowtriangle$.

Behavioural semantics One of the main motivation of our work is the definition of a labelled bisimilarity for MA. Rather than simply defining an ad-hoc bisimulation based equivalence over systems we first introduce our reference equivalence: reduction barbed congruence.

Definition 1.1 *A relation $\mathcal{R}$ over processes is* reduction closed *if $P\ \mathcal{R}\ Q$ and $P \rightarrowtriangle P'$ imply the existence of some Q' such that $Q \rightarrowtriangle^* Q'$ and $P'\ \mathcal{R}\ Q'$.*

Definition 1.2 *A relation $\mathcal{R}$ over processes is* preserved by contexts (*resp.* system contexts) *if $P\ \mathcal{R}\ Q$ implies $C[P]\ \mathcal{R}\ C[Q]$ for all contexts (resp. system contexts) $C[-]$.*

In Mobile Ambients, given a process P, a simple observable is the presence at top-level of an ambient whose name (say n) is not restricted: the observation predicate $P \downarrow n$ captures exactly this observable. Formally, we write $P \downarrow n$ if $P \equiv (\boldsymbol{\nu}\tilde{m})(n[P_1] \mid P_2)$ where $n \notin \{\tilde{m}\}$. We write $P \Downarrow n$ if there exists P' such that $P \rightarrowtriangle^* P'$ and $P' \downarrow n$.

Definition 1.3 *We say that a relation $\mathcal{R}$ over processes is* barb preserving *if $P\ \mathcal{R}\ Q$ and $P \downarrow n$ implies $Q \Downarrow n$.*

We are ready to define our contextual equivalences:

Table 2 Structural Congruence and Reduction Rules

$P \mid Q \equiv P \mid Q$	(Struct Par Comm)
$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$	(Struct Par Assoc)
$P \mid \mathbf{0} \equiv P$	(Struct Zero Par)
$(\boldsymbol{\nu} n)\mathbf{0} \equiv \mathbf{0}$	(Struct Zero Res)
$!C.P \equiv C.P \mid !C.P$	(Struct Repl Par)
$(\boldsymbol{\nu} n)(\boldsymbol{\nu} m)P \equiv (\boldsymbol{\nu} m)(\boldsymbol{\nu} n)P$	(Struct Res Res)
$n \notin \mathrm{fn}(P)$ implies $(\boldsymbol{\nu} n)(P \mid Q) \equiv P \mid (\boldsymbol{\nu} n)Q$	(Struct Res Par)
$n \neq m$ implies $(\boldsymbol{\nu} n)(m[P]) \equiv m[(\boldsymbol{\nu} n)P]$	(Struct Res Amb)

$\equiv$ is the least equivalence relation which satisfies the axioms and rules above, and is preserved by contexts.

$n[\mathtt{in}\_m.P \mid Q] \mid m[R] \rightarrow m[\,n[P \mid Q] \mid R\,]$	(Red In)
$m[\,n[\mathtt{out}\_m.P \mid Q] \mid R\,] \rightarrow n[P \mid Q] \mid m[R]$	(Red Out)
$\mathtt{open}\_n.P \mid n[\,Q\,] \rightarrow P \mid Q$	(Red Open)
$P \equiv Q \quad Q \rightarrow R \quad R \equiv S$ implies $P \rightarrow S$	(Red Struct)

$\rightarrow$ is the least relation which satisfies the rules above and is preserved by static contexts.

Definition 1.4 (Reduction barbed congruence)

- Reduction barbed congruence over systems, *written* $\cong_{\mathrm{s}}$, *is the largest symmetric relation over systems which is reduction closed, barb preserving, and preserved by system contexts.*
- Reduction barbed congruence over processes, *written* $\cong_{\mathrm{p}}$, *is the largest symmetric relation over processes which is reduction closed, barb preserving, and preserved by all contexts.*

In the remainder of the paper, when working with a relation $\mathcal{R}$ over processes and/or systems, we write $\mathcal{R}^{=}$ to denote the symmetric closure of $\mathcal{R}$.

2 A Labelled Transition Semantics for Systems

Along standard lines, [25], prefixes C give rise to transitions of the form $P \xrightarrow{C} Q$. For example we have

$$\mathtt{in}\_n.P_1 \mid P_2 \xrightarrow{\mathtt{in}\_n} P_1 \mid P_2.$$

However, similarly to what happens in [23] each of the capability C induces different and more complicated actions. The LTS is defined over processes, although in the labelled bisimilarity we only consider actions going from systems to systems. We make a distinction between *pre-actions* and *env-actions*: the former denote the possibility to exercise certain capabilities whereas the latter model the interaction of a system with its environment. As usual, we also have *τ-actions* to model internal computations. Only env-actions and τ-actions model the evolution of a system at run-time.

The pre-actions, defined in Table 4, are of the form $P \xrightarrow{\pi} O$ where the ranges of π and of O, the *outcomes*, are reported in Table 3. An outcome may be a simple process Q, if for example π

Table 3 Pre-actions, Env-actions, Actions, Concretions, and Outcomes

Pre-actions:	π	::=		*Outcomes:*	O	::=	$P \mid K$
		$\mid$	$\texttt{in\_}n \mid \texttt{out\_}n$				
		$\mid$	$\texttt{open\_}n \mid \texttt{enter\_}n$				
		$\mid$	$\texttt{amb\_}n \mid \texttt{exit\_}n$				
Env-actions:	σ	::=		*Concretions:*	K	::=	$(\boldsymbol{\nu}\tilde{m})\langle P\rangle Q$
		$\mid$	$k.\texttt{enter\_}n \mid k.\texttt{exit\_}n$				
		$\mid$	$*.\texttt{enter\_}n \mid *.\texttt{exit\_}n$				
		$\mid$	$n.\overline{\texttt{enter}}\texttt{\_}k \mid k.\texttt{open\_}n$				
Actions:	α	::=	$\sigma \mid \tau$				

is a prefix of the language, or a *concretion*, of the form $(\boldsymbol{\nu}\tilde{m})\langle P\rangle Q$, when an ambient boundary is somehow involved. In this case, P represents the code that may enter to, reside at, or exit from an ambient; Q represents the derivative which is not affected by the action, and $\tilde{m}$ is the set of private names shared by P and Q. We adopt the convention that if K is the concretion $(\boldsymbol{\nu}\tilde{m})\langle P\rangle Q$, then $(\boldsymbol{\nu}r)K$ is a shorthand for $(\boldsymbol{\nu}\tilde{m})\langle P\rangle(\boldsymbol{\nu}r)Q$, if $r \notin \mathrm{fn}(P)$, and the concretion $(\boldsymbol{\nu}r\tilde{m})\langle P\rangle Q$ otherwise. We have a similar convention for the rule (π Par): $K \mid R$ is defined to be the concretion $(\boldsymbol{\nu}\tilde{m})\langle P\rangle(Q \mid R)$, where $\tilde{m}$ are chosen, using α-conversion if necessary, so that $\mathrm{fn}(R) \cap \{\tilde{m}\} = \emptyset$; similarly $R \mid K$ is the concretion $(\boldsymbol{\nu}\tilde{m})\langle P\rangle(R \mid Q)$. Occasionally, we omit inactive processes when they are in parallel with processes, writing P for $P \mid \mathbf{0}$.

The rules (π Pfx), (π Repl Pfx), (π Res), and (π Par) are standard. The rule (π Enter) results in a concretion containing the ambient willing to enter n. The rule (π Exit) is similar, but the resulting concretion contains the ambient willing to exit from n. The rule (π Amb) records in a concretion the code residing at n.

The τ-actions, formally defined in Table 5, model the internal evolution of processes. The rule (π Enter) models an ambient migrating into a sibling ambient n. The rule (π Exit) models an ambient k exiting from an ambient n. The rule (π Open) describes the opening of an ambient n. Structural rules (π Amb), (π Res), and (π Par) are straightforward.

The env-actions, formally defined in Table 6, are of the form $M \xrightarrow{\sigma} M'$, where the range of σ is given in Table 3. Env-actions turn concretions into running systems by explicitly introducing the environment's ambient interacting with the process in question. The content of this ambient will be instantiated later, in the definition of the bisimilarity, with a process. For convenience, *we extend the syntax of processes with the special process* $\circ$ to pinpoint those ambients whose content will be instantiated later. The process $\circ$ does not reduce, and, from an operational point of view, it can be assimilated to the inactive process: it is simply a placeholder. Notice that, unlike pre-actions and τ-actions, env-actions do not have structural rules; this is because env-actions are supposed to be performed by systems that can directly interact with the environment.

In the rules (Enter) and (Exit) an ambient k enters, respectively exit from, an ambient n provided by the environment. The rules (Enter Shh) and (Exit Shh) are similar and model the migration of private ambients. In the rule (Co-Enter) an ambient k, provided by the environment, migrates into an ambient n of the process. In the rule (Open) the environment opens

Table 4 Labelled Transition System - Pre-actions

$$(\pi \text{ Pfx}) \ \frac{-}{\pi.P \xrightarrow{\pi} P} \qquad (\pi \text{ Repl Pfx}) \ \frac{-}{!\pi.P \xrightarrow{\pi} P \mid !\pi.P}$$

$$(\pi \text{ Enter}) \ \frac{P \xrightarrow{\texttt{in\_}n} P_1}{m[P] \xrightarrow{\texttt{enter\_}n} \langle m[P_1]\rangle \mathbf{0}} \qquad (\pi \text{ Amb}) \ \frac{-}{n[P] \xrightarrow{\texttt{amb\_}n} \langle P\rangle \mathbf{0}}$$

$$(\pi \text{ Exit}) \ \frac{P \xrightarrow{\texttt{out\_}n} P_1}{m[P] \xrightarrow{\texttt{exit\_}n} \langle m[P_1]\rangle \mathbf{0}} \qquad (\pi \text{ Res}) \ \frac{P \xrightarrow{\pi} O \qquad n \notin \text{fn}(\pi)}{(\boldsymbol{\nu} n)P \xrightarrow{\pi} (\boldsymbol{\nu} n)O}$$

$$(\pi \text{ Par}) \ \frac{P \xrightarrow{\pi} O}{\begin{array}{c} P \mid Q \xrightarrow{\pi} O \mid Q \\ Q \mid P \xrightarrow{\pi} Q \mid O \end{array}}$$

Table 5 Labelled Transition System - τ-actions

$$(\tau \text{ Enter}) \ \frac{P \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{p})\langle P_1\rangle P_2 \quad Q \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{q})\langle Q_1\rangle Q_2 \ ^{(*)}}{\begin{array}{c} P \mid Q \xrightarrow{\tau} (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{q})(n[P_1 \mid Q_1] \mid P_2 \mid Q_2) \\ Q \mid P \xrightarrow{\tau} (\boldsymbol{\nu}\tilde{q})(\boldsymbol{\nu}\tilde{p})(n[Q_1 \mid P_1] \mid Q_2 \mid P_2) \end{array}}$$

$$(\tau \text{ Exit}) \ \frac{P \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{m})\langle k[P_1]\rangle P_2}{n[P] \xrightarrow{\tau} (\boldsymbol{\nu}\tilde{m})(k[P_1] \mid n[P_2])} \qquad (\tau \text{ Amb}) \ \frac{P \xrightarrow{\tau} Q}{n[P] \xrightarrow{\tau} n[Q]}$$

$$(\tau \text{ Open}) \ \frac{P \xrightarrow{\texttt{open\_}n} P_1 \quad Q \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{m})\langle Q_1\rangle Q_2}{\begin{array}{c} P \mid Q \xrightarrow{\tau} P_1 \mid (\boldsymbol{\nu}\tilde{m})(Q_1 \mid Q_2) \\ Q \mid P \xrightarrow{\tau} (\boldsymbol{\nu}\tilde{m})(Q_1 \mid Q_2) \mid P_1 \end{array}} \qquad (\tau \text{ Res}) \ \frac{P \xrightarrow{\tau} P'}{(\boldsymbol{\nu} n)P \xrightarrow{\tau} (\boldsymbol{\nu} n)P'}$$

$$(\tau \text{ Par}) \ \frac{P \xrightarrow{\tau} P'}{\begin{array}{c} P \mid Q \xrightarrow{\tau} P' \mid Q \\ Q \mid P \xrightarrow{\tau} Q \mid P' \end{array}}$$

(*) In rule (τ Enter) we require $((\text{fn}(P_1) \cup \text{fn}(P_2)) \cap \{\tilde{q}\}) = ((\text{fn}(Q_1) \cup \text{fn}(Q_2)) \cap \{\tilde{p}\}) = \emptyset$.

an ambient n of the process; the opening is performed inside an ambient k provided by the environment.

We call *actions* the set of env-actions extended with τ. Actions, denoted by α, always go from systems to systems and, in general, from processes to processes, even if the outcome may possibly involve the special process $\circ$. As our bisimilarity will be defined over systems, we will only consider actions (and not pre-actions) in its definition.

Proposition 2.1 *If T is a system (resp. a process), and $T \xrightarrow{\alpha} T'$, then T' is a system (resp. a process), possibly containing the special process $\circ$.*

Table 6 Labelled Transition System - Env-actions

$$\text{(Enter)}\ \frac{P \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{m})\langle k[P_1]\rangle P_2 \qquad k \notin \tilde{m}}{P \xrightarrow{k.\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{m})(n[k[P_1] \mid \circ] \mid P_2)}$$

$$\text{(Co-Enter)}\ \frac{P \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{m})\langle P_1\rangle P_2 \qquad k \notin \tilde{m}}{P \xrightarrow{n.\overline{\texttt{enter}}\_k} (\boldsymbol{\nu}\tilde{m})(n[P_1 \mid k[\,\circ\,]] \mid P_2)}$$

$$\text{(Exit)}\ \frac{P \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{m})\langle k[P_1]\rangle P_2 \qquad k \notin \tilde{m}}{P \xrightarrow{k.\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{m})(k[P_1] \mid n[\,\circ \mid P_2])}$$

$$\text{(Open)}\ \frac{P \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{m})\langle P_1\rangle P_2}{P \xrightarrow{k.\texttt{open\_}n} k[\circ \mid (\boldsymbol{\nu}\tilde{m})(P_1 \mid P_2)]}$$

$$\text{(Enter Shh)}\ \frac{P \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{m})\langle k[P_1]\rangle P_2 \qquad k \neq n \qquad k \in \tilde{m}}{P \xrightarrow{*.\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{m})(n[k[P_1] \mid \circ] \mid P_2)}$$

$$\text{(Exit Shh)}\ \frac{P \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{m})\langle k[P_1]\rangle P_2 \qquad k \neq n \qquad k \in \tilde{m}}{P \xrightarrow{*.\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{m})(k[P_1] \mid n[\circ \mid P_2])}$$

Since we are interested in *weak bisimilarities*, that abstract over τ-actions, we introduce the notion of weak action. The definition is standard: $\Rightarrow$ denotes the reflexive and transitive closure of $\xrightarrow{\tau}$; $\stackrel{\alpha}{\Longrightarrow}$ denotes $\Rightarrow\xrightarrow{\alpha}\Rightarrow$; $\stackrel{\hat{\alpha}}{\Longrightarrow}$ denotes $\Rightarrow$ if $\alpha = \tau$ and $\stackrel{\alpha}{\Longrightarrow}$ otherwise.

Now, let us explain with an example the rules induced by the prefix `in`, the *immigration* of ambients. A typical example of an ambient m migrating into an ambient n follows:

$$(\boldsymbol{\nu}m)(m[\,\texttt{in\_}n.P_1 \mid P_2\,] \mid M) \mid n[Q] \rightarrowtriangle (\boldsymbol{\nu}m)(M \mid n[\,m[\,P_1 \mid P_2\,] \mid Q])$$

The driving force behind the migration is the activation of the prefix $\texttt{in\_}n$, within the ambient m. It induces a capability in the ambient m to migrate into n, that we formalise as a new action $\texttt{enter\_}n$. Thus, an application of (π Enter) gives

$$m[\texttt{in\_}n.P_1 \mid P_2] \xrightarrow{\texttt{enter\_}n} \langle m[P_1 \mid P_2]\rangle \mathbf{0}$$

and, more generally, using the structural rules (π Res) and (π Par),

$$(\boldsymbol{\nu}m)(m[\texttt{in\_}n.P_1 \mid P_2] \mid M) \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}m)\langle m[P_1 \mid P_2]\rangle M.$$

This means that the ambient $m[\texttt{in\_}n.P_1 \mid P_2]$ has the capability to enter an ambient n; if the capability is exercised, the ambient $m[P_1 \mid P_2]$ will enter n while M will be the residual where the execution started. Of course the action can only be realised if there is an ambient n in parallel. The rule (π Amb) allows to check for the presence of ambients. So for example, we have

$$n[Q] \xrightarrow{\texttt{amb\_}n} \langle Q\rangle \mathbf{0}.$$

Here, the concretion $\langle Q \rangle \mathbf{0}$ says that the process Q is inside n and is affected by the action, while the process $\mathbf{0}$ is outside and is not affected. Finally, the rule (τ Enter) allows these two complementary actions to occur simultaneously, executing the migration of the ambient $m[P_1 \mid P_2]$ from its current computation space into the ambient n, giving rise to the original move above:

$$(\boldsymbol{\nu} m)(m[\,\mathtt{in}\_n.P_1 \mid P_2\,] \mid M) \mid n[Q] \xrightarrow{\tau} (\boldsymbol{\nu} m)(M \mid n[\,m[\,P_1 \mid P_2\,] \mid Q]).$$

Note that this is a *higher-order* interaction, as the ambient $m[P_1 \mid P_2]$ is transferred between two computation spaces.

We have not said yet what env-actions are useful for. They model the interaction of mobile agents with their environment. So, for instance, using the rule (Enter Shh), we derive from

$$(\boldsymbol{\nu} m)(m[\mathtt{in}\_n.P_1 \mid P_2] \mid M) \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu} m)\langle m[P_1 \mid P_2]\rangle M.$$

the transition

$$(\boldsymbol{\nu} m)(m[\mathtt{in}\_n.P_1 \mid P_2] \mid M) \xrightarrow{*.\mathtt{enter}\_n} (\boldsymbol{\nu} m)(n[m[P_1 \mid P_2] \mid \circ] \mid M).$$

This transition denotes a private (*secret*) ambient entering an ambient n provided by the environment. The computation running at n will be added later by instantiating the placeholder $\circ$.

Had the ambient name m not been restricted, we would have used the rule (Enter) to derive

$$m[\mathtt{in}\_n.P_1 \mid P_2] \mid M \xrightarrow{m.\mathtt{enter}\_n} n[m[P_1 \mid P_2] \mid \circ] \mid M$$

to model a global ambient m entering an ambient n provided by the environment.

Whenever a system offers a public ambient n at top-level, a context can interact with the system by providing an ambient entering n. The rule (Co-Enter) captures this interaction between system and environment.

Now, let us explain the rules for *emigration* with an example. A typical example of an ambient m emigrating from an ambient n is as follows:

$$n[\,m[\mathtt{out}\_n.P_1 \mid P_2] \mid Q] \rightarrowtriangle m[P_1 \mid P_2] \mid n[Q].$$

The driving force behind the emigration is the activation of the prefix $\mathtt{out}\_n$ within the ambient m. It induces a capability in the ambient m to emigrate from n, which we formalise as a new action $\mathtt{exit}\_n$. Thus an application of the rule (π Exit), followed by (π Par), gives

$$m[\mathtt{out}\_n.P_1 \mid P_2] \mid Q \xrightarrow{\mathtt{exit}\_n} \langle m[P_1 \mid P_2]\rangle Q.$$

Here, when exercising this capability, the code Q remains inside the ambient n while the ambient $m[P_1 \mid P_2]$ moves outside. However, to complete the emigration of m we need a further context, namely the ambient n from which to emigrate. This leads to the rule (τ Exit); an application of which gives the original move above:

$$n[\,m[\mathtt{out}\_n.P_1 \mid P_2] \mid Q] \xrightarrow{\tau} m[P_1 \mid P_2] \mid n[Q].$$

As for immigration, env-actions $m.\mathtt{exit}\_n$ and $*.\mathtt{exit}\_n$ model the exiting of global and provate ambients from an ambient n provided by the environment.

The rule that controls the *opening* is straightforward and left to reader.

We end this section with a theorem that asserts that the LTS-based semantics coincides with the reduction semantics of Section 1.

For any process P, outcome O and pre-action π such that $P \xrightarrow{\pi} O$, the structure of P and O can be determined up to structural congruence.

Lemma 2.2

- *If* $P \xrightarrow{C} O$, *with* $C \in \{\mathtt{in}\_n, \mathtt{out}\_n, \mathtt{open}\_n\}$, *then there exist* $\tilde{p}, P_1, P_2$, *with* $n \notin \tilde{p}$, *such that*
$$P \equiv (\boldsymbol{\nu}\tilde{p})(C.P_1 \mid P_2) \quad \textit{and} \quad O \equiv (\boldsymbol{\nu}\tilde{p})(P_1 \mid P_2)\ .$$
- *If* $P \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{p})\langle P'\rangle P''$ *then there exist* k, P_1, P_2, *with* $n \notin \tilde{p}$, *such that*
$$P \equiv (\boldsymbol{\nu}\tilde{p})(k[\mathtt{in}\_n.P_1 \mid P_2] \mid P'') \quad \textit{and} \quad P' \equiv k[P_1 \mid P_2]\ .$$
- *If* $P \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{p})\langle P'\rangle P''$ *then there exist* k, P_1, P_2, *with* $n \notin \tilde{p}$, *such that*
$$P \equiv (\boldsymbol{\nu}\tilde{p})(k[\mathtt{out}\_n.P_1 \mid P_2] \mid P'') \quad \textit{and} \quad P' \equiv k[P_1 \mid P_2]\ .$$
- *If* $P \xrightarrow{\mathtt{amb}\_n} (\boldsymbol{\nu}\tilde{p})\langle P'\rangle P''$, *with* $n \notin \tilde{p}$, *then* $P \equiv (\boldsymbol{\nu}\tilde{p})(n[P'] \mid P'')$.

Proof By induction on the transition rules of Tables 4 and 5. □

Theorem 2.3

1. *If* $P \xrightarrow{\tau} P'$ *then* $P \rightarrow\!\!\triangleright P'$
2. *If* $P \rightarrow\!\!\triangleright P'$ *then* $P \xrightarrow{\tau} \equiv P'$.

The proof is standard, and is reported in Appendix A. An easy consequence of this result is that structural congruence is preserved by τ-actions.

Corollary 2.4 *If* $M \equiv N$ *and* $M \xrightarrow{\tau} M'$, *then there is* N' *such that* $N \xrightarrow{\tau} N'$ *and* $M' \equiv N'$.

3 Characterising Reduction Barbed Congruence over Systems

In this section we define a labelled bisimilarity that completely characterises reduction barbed congruence over systems.

In the previous section we said that env-actions introduce a special process $\circ$ to pinpoint those ambients whose content will be instantiated in the bisimilarity. This means that we will have an operator $\bullet$ to instantiate the placeholder with a process.

Definition 3.1 *Let P and Q be processes which may contain instances of $\circ$. Let R be a process. We define:*

$$\begin{array}{rcl@{\qquad}rcl} \mathbf{0} \bullet R & \stackrel{\text{def}}{=} & \mathbf{0} & (P \mid Q) \bullet R & \stackrel{\text{def}}{=} & (P \bullet R) \mid (Q \bullet R) \\ n[P] \bullet R & \stackrel{\text{def}}{=} & n[P \bullet R] & (\boldsymbol{\nu} n)P \bullet R & \stackrel{\text{def}}{=} & (\boldsymbol{\nu} n)(P \bullet R) \text{ if } n \notin \text{fn}(R) \\ \circ \bullet R & \stackrel{\text{def}}{=} & R & C.P \bullet R & \stackrel{\text{def}}{=} & C.(P \bullet R) \\ !C.P \bullet R & \stackrel{\text{def}}{=} & !C.(P \bullet R). & & & \end{array}$$

The $\bullet$ operator performs a name-capture avoiding substitution. It should be pointed out that we allow structural congruence to rearrange terms containing $\circ$: with respect to structural congruence, $\circ$ behaves like the inactive process $\mathbf{0}$. This motivates the introduction of the special process $\circ$, instead of relying on standard process substitutions. In some proofs, we will refer to an extended definition of $\bullet$ allowing also R to range over processes with $\circ$.

Everything is now in place to define our bisimilarity.

Definition 3.2 (Late bisimilarity) *A symmetric relation $\mathcal{R}$ over systems is a* late bisimulation *if $M \mathcal{R} N$ implies:*

- *if $M \xrightarrow{\alpha} M'$, $\alpha \notin \{*.\texttt{enter}\_n, *.\texttt{exit}\_n\}$, then there is a system N' such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} N'$ and for all processes P it holds $M' \bullet P \mathcal{R} N' \bullet P$;*
- *if $M \xrightarrow{*.\texttt{enter}\_n} M'$ then there is a system N' such that $N \mid n[\circ] \Rightarrow N'$ and for all processes P it holds $M' \bullet P \mathcal{R} N' \bullet P$;*
- *if $M \xrightarrow{*.\texttt{exit}\_n} M'$ then there is a system N' such that $n[\circ \mid N] \Rightarrow N'$ and for all processes P it holds $M' \bullet P \mathcal{R} N' \bullet P$.*

Systems M and N are late bisimilar*, written $M \approx N$, if $M \mathcal{R} N$ for some late bisimulation $\mathcal{R}$.*

Some comments. When $\alpha = \tau$, the derivative M' does not contain the special process $\circ$, as there is no interaction with the environment. As a consequence, for $\alpha = \tau$, we could simply write

- if $M \xrightarrow{\tau} M'$ then there is a system N' such that $N \Rightarrow N'$ and $M' \mathcal{R} N'$.

On the other hand, when α is an env-action, there is a universal quantification over the process P provided by the environment. This process instantiates the placeholder $\circ$ generated by the env-action.

The bisimilarity is defined in a *late* style as the existential quantification precedes the universal one. Another possibility would be to define the bisimilarity in *early* style, where the universal quantification over the environment's contribution P precedes that over the derivative N'. We write $\approx_e$ to denote the early variant. By definition, every late bisimulation is also a early one, while the converse, in general, does not hold. However, in our case, as in HOπ [31], we will prove that late and early bisimilarity coincide. We choose late bisimilarity as our main labelled bisimilarity because the derivatives N' do not depend on the environment's contribution P.

Finally, the π-calculus experience suggests that late bisimulations may fail to be transitive. However, processes reveals to be more 'tractable' than names, and in our framework late bisimilarity turns out to be an equivalence relation.

Remark 3.3 (Invisible actions) *The reader may wonder why the bisimulation is not defined in the standard way, as a symmetric relation $\mathcal{R}$ over systems such that whenever $M\ \mathcal{R}\ N$ and $M \xrightarrow{\alpha} M'$, there is a system N' such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} N'$, and for all processes P it holds that $M' \bullet P\ \mathcal{R}\ N' \bullet P$. While this equivalence is a sound proof technique, it is not a complete characterisation of $\cong_{\mathrm{s}}$. In fact, the two systems*

$$(\boldsymbol{\nu} n)n[\mathtt{in}\_k.\mathbf{0}] \quad \textit{and} \quad \mathbf{0}$$

are reduction barbed congruent, but are distinguished by the equivalence defined above. The system $(\boldsymbol{\nu} n)n[\mathtt{in}\_k.\mathbf{0}]$ can perform a $*.\mathtt{enter}\_k$ action while $\mathbf{0}$ cannot. This example shows that a labelled characterisation of reduction barbed congruence should treat actions $*.\mathtt{enter}\_n$ and $*.\mathtt{exit}\_n$ separately, asking for weaker matching requirements: like input actions in the asynchronous π-calculus [16, 3], these actions cannot be observed by a context.

3.1 Soundness

Here, we show that late and early bisimilarity are two proof techniques for reduction barbed congruence over systems. More precisely we prove that they are both contained in reduction barbed congruence over systems.

The following lemma is crucial for proving that $\approx$ is preserved by system contexts. This lemma will be also used for proving the soundness of the up-to context proof technique in Section 4.

Lemma 3.4 *Let $\mathcal{S}$ be a symmetric relation between systems preserved by system contexts. Let $(M, N) \in \mathcal{S}$ be a pair satisfying the bisimulation conditions in $\mathcal{S}$, that is,*

- *if $M \xrightarrow{\alpha} M'$, $\alpha \notin \{*.\mathtt{enter}\_n, *.\mathtt{exit}\_n\}$, then there is a system N' such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} N'$ and for all processes P it holds $M' \bullet P\ \mathcal{S}\ N' \bullet P$;*
- *if $M \xrightarrow{*.\mathtt{enter}\_n} M'$ then there is a system N' such that $N \mid n[\,\circ\,] \Rightarrow N'$ and for all processes P it holds $M' \bullet P\ \mathcal{S}\ N' \bullet P$;*
- *if $M \xrightarrow{*.\mathtt{exit}\_n} M'$ then there is a system N' such that $n[\circ \mid N] \Rightarrow N'$ and for all processes P it holds $M' \bullet P\ \mathcal{S}\ N' \bullet P$.*

Then, all the pairs $(C[M], C[N])$, for any system context $C[-]$, also satisfy the bisimulation conditions in $\mathcal{S}$.

The proof is by induction over the structure of $C[-]$, and is reported in Appendix B.

Theorem 3.5 *Late bisimilarity is preserved by system contexts.*

Proof Let $\mathcal{S}$ be the smallest binary relation between systems such that:

1. $\approx \subseteq \mathcal{S}$;
2. if $M\ \mathcal{S}\ N$, then $C[M]\ \mathcal{S}\ C[N]$ for all system contexts $C[-]$.

Remark that $\mathcal{S}$ is symmetric because of the symmetry of $\approx$. We prove that $\mathcal{S}$ is a late bisimilarity up to $\equiv$[2], by induction on the definition of $\mathcal{S}$. This is sufficient to conclude that $\approx$ is preserved by system contexts.

[2]The soundness of the up to $\equiv$ proof technique follows easily from Corollary 2.4.

Table 7 System Contexts for Visible Actions

$$
\begin{aligned}
C_{k.\mathtt{enter}\_n}[-] &\stackrel{\text{def}}{=} n[\mathsf{done}[\mathtt{in}\_k.\mathtt{out}\_k.\mathtt{out}\_n] \mid \circ] \mid - \\
C_{k.\mathtt{exit}\_n}[-] &\stackrel{\text{def}}{=} (\boldsymbol{\nu} a)a[\mathtt{in}\_k.\mathtt{out}\_k.\mathsf{done}[\mathtt{out}\_a]] \mid n[\circ \mid -] \\
C_{n.\overline{\mathtt{enter}}\_k}[-] &\stackrel{\text{def}}{=} (\boldsymbol{\nu} a)a[\mathtt{in}\_n.k[\mathtt{out}\_a.(\circ \mid (\boldsymbol{\nu} b)b[\mathtt{out}\_k.\mathtt{out}\_n.\mathsf{done}[\mathtt{out}\_b]])]] \mid - \\
C_{k.\mathtt{open}\_n}[-] &\stackrel{\text{def}}{=} k[\circ \mid (\boldsymbol{\nu} a, b)(\mathtt{open}\_b.\mathtt{open}\_a.\mathsf{done}[\mathtt{out}\_k] \mid a[- \mid \mathtt{open}\_n.b[\mathtt{out}\_a]])]
\end{aligned}
$$

where a,b and done are fresh names.

- $M \mathcal{S} N$ because $M \approx N$. Immediate.
- $C[M] \mathcal{S} C[N]$ because $M \mathcal{S} N$.

  The induction hypothesis assures that $(M, N) \in \mathcal{S}$ is a pair satisfying the bisimulation conditions in $\mathcal{S}$. Lemma 3.4 assures that the pair $(C[M], C[N])$ satisfies the bisimulation conditions in $\mathcal{S}$. □

It is easy to adapt Lemma 3.4 and the above proof to show that also early bisimilarity is preserved by system contexts.

Proposition 3.6 *Early bisimilarity is preserved by system contexts.*

In the following lemma we point out a close relationship between the observation predicate $M \downarrow n$ and a specific action that M can emit.

Lemma 3.7

1. *If* $M \xrightarrow{n.\overline{\mathtt{enter}}\_k} M'$ *then* $M \downarrow n$;
2. *if* $M \downarrow n$ *then there exists a system* M' *such that* $M \xrightarrow{n.\overline{\mathtt{enter}}\_k} M'$, *for some* k.

Now we can prove that both late and early bisimilarity are contained in the reduction barbed congruence over systems.

Theorem 3.8 (Soundness) *The following chain of inclusions hold* $\approx \subseteq \approx_e \subseteq \cong_{\mathrm{s}}$.

Proof The first inclusion holds by definition. The second one comes from the fact that early bisimilarity is reduction closed (by Theorem 2.3, part 1), barb-preserving (by Lemma 3.7), and preserved by system contexts (by Proposition 3.6). □

3.2 Completeness

We now prove that late and early bisimilarity are more than proof techniques. They actually characterise reduction barbed congruence over systems. The main challenge here is to design the system contexts capable to observe our visible actions.

The definition of these contexts, $C_\alpha[-]$, for every visible action α, is given in Table 7. The ambient done is used as *fresh* barb to signal the consumption of the actions. In the context for $k.\mathtt{enter}\_n$ the ambient done is used as a pilot ambient to verify the ambient's move. The

context for $k.\texttt{exit\_}n$ uses a private ambient a, different from done, as the pilot ambient. This is because the barb done must be unleashed only after the exit move has been performed. The context for $n.\overline{\texttt{enter}}\texttt{\_}k$ is more subtle. Instead of moving directly k into n we encapsulate k inside a private ambient a to avoid interferences. More precisely, to prevent the ambient k is used to enter into the ambient n by a Trojan horse hidden in the system plugged into the hole (this could invalidate Lemma 3.14). Only after k has reached n we release a barb done. The private ambient b assures that if the ambient done arrives at top-level, then it is empty. This allows a uniform formulation of Lemma 3.14. Finally, in the context for $k.\texttt{open\_}n$, the private ambients a and b guarantee that the barb done is unleashed only after the opening of ambient n.

To prove our characterisation result we will show that reduction barbed congruence over systems is contained in the late bisimilarity. Then, by Theorem 3.8, we can prove that late bisimulation, early bisimulation, and reduction barbed congruence over systems, they all coincide. To prove that reduction barbed congruence over systems implies late bisimilarity we must spell out the correspondence between visible actions α and their corresponding system contexts $C_\alpha[-]$.

The following lemma says that the distinguishing system contexts of Table 7 are sound, that is, they can successfully mimic the execution of visible actions.

Lemma 3.9 *Let M be a system. Let $\alpha \in \{k.\texttt{enter\_}n, k.\texttt{exit\_}n, n.\overline{\texttt{enter}}\texttt{\_}k, k.\texttt{open\_}n\}$. For all processes P, if $M \xrightarrow{\alpha} M'$ then $C_\alpha[M] \bullet P \Rightarrow\cong_{\mathrm{s}} (M' \bullet P) \mid \mathsf{done}[\,]$.*

Proof The proof is by case analysis on α.

Case $\alpha = k.\texttt{enter\_}n$. Let P be a process. We know that $M \xrightarrow{k.\texttt{enter}\_n} M'$. Then

$$M \equiv (\boldsymbol{\nu}\tilde{m})(k[\texttt{in\_}n.M_1 \mid M_2] \mid M_3)$$

where $(\{n,k\} \cup \mathrm{fn}(P)) \cap \{\tilde{m}\} = \emptyset$, and

$$M' \equiv (\boldsymbol{\nu}\tilde{m})(n[k[M_1 \mid M_2] \mid \circ] \mid M_3).$$

Now,

$$\begin{array}{rl}
 & C_{k.\texttt{enter}_n}[M] \bullet P \\
\equiv & (\boldsymbol{\nu}\tilde{m})(n[\mathsf{done}[\texttt{in_}k.\texttt{out_}k.\texttt{out_}n] \mid P] \mid k[\texttt{in_}n.M_1 \mid M_2] \mid M_3) \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[\mathsf{done}[\texttt{in_}k.\texttt{out_}k.\texttt{out_}n] \mid P \mid k[M_1 \mid M_2]] \mid M_3) \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[P \mid k[M_1 \mid M_2 \mid \mathsf{done}[\texttt{out_}k.\texttt{out_}n]]] \mid M_3) \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[P \mid \mathsf{done}[\texttt{out_}n] \mid k[M_1 \mid M_2]] \mid M_3) \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(\mathsf{done}[\,] \mid n[P \mid k[M_1 \mid M_2]] \mid M_3) \\
\equiv & (\boldsymbol{\nu}\tilde{m})(n[\circ \mid k[M_1 \mid M_2] \mid M_3]) \bullet P \mid \mathsf{done}[\,] \\
= & M' \bullet P \mid \mathsf{done}[\,]
\end{array}$$

By Corollary 2.4 and transitivity of $\equiv$, there exists a system O such that $C_{k.\texttt{enter}\_n}[M] \bullet P \Rightarrow O$, and $O \equiv M' \bullet P \mid \mathsf{done}[\,]$. The result follows because structural congruence restricted to systems is contained in reduction barbed congruence over systems, and $O \cong_{\mathrm{s}} M' \bullet P \mid \mathsf{done}[\,]$.

The remaining cases can be found in the Appendix B. □

Table 8 Spy Contexts

$$\mathsf{spy}_\alpha\langle i,j,-\rangle \stackrel{\text{def}}{=} (i[\texttt{out\_}n] \mid -) \oplus (j[\texttt{out\_}n] \mid -) \quad \text{if } \alpha \in \{k.\texttt{enter\_}n, k.\texttt{exit\_}n, k.\texttt{open\_}n, *.\texttt{enter\_}n, *.\texttt{exit\_}n\}$$

$$\mathsf{spy}_\alpha\langle i,j,-\rangle \stackrel{\text{def}}{=} (i[\texttt{out\_}k.\texttt{out\_}n] \mid -) \oplus (j[\texttt{out\_}k.\texttt{out\_}n] \mid -) \text{ if } \alpha \in \{n.\overline{\texttt{enter}}\_k\}$$

To complete the correspondence proof between actions α and their contexts $C_\alpha[-]$, we have to prove the converse of Lemma 3.9, formalised in Lemma 3.14. The proof of this result uses some special contexts $\mathsf{spy}_\alpha\langle i,j,-\rangle$, defined in Table 8, as a technical tool to guarantee that the process P provided by the environment does not perform any action. This is necessary when proving completeness to guarantee that the contribution P is the same on both sides. Formally, the $\mathsf{spy}_\alpha\langle i,j,-\rangle$ contexts are *multi-hole contexts* [36] as the same hole occurs more than once (in this case, twice). The $\mathsf{spy}_\alpha\langle i,j,-\rangle$ contexts use *internal choice* encoded as:

$$P \oplus Q \stackrel{\text{def}}{=} (\boldsymbol{\nu} o)(o[\,] \mid \texttt{open\_}o.P \mid \texttt{open\_}o.Q) \ .$$

This encoding satisfies the following properties:

Lemma 3.10 *$P \oplus Q \xrightarrow{\tau} \cong_{\mathrm{s}} P$ and $P \oplus Q \xrightarrow{\tau} \cong_{\mathrm{s}} Q$.*

The ability of $\mathsf{spy}_\alpha\langle i,j,P\rangle$ to 'spy' on P stems from the fact that one of the two fresh barbs i and j is lost when P performs any action. The key properties of $\mathsf{spy}_\alpha\langle i,j,-\rangle$ are captured by the lemma below, proved in Appendix B.

Lemma 3.11

1. *Let M be a system which may possibly contain an occurrence of the special process $\circ$. If $M \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \xrightarrow{\tau} O$ and $O \Downarrow_{i,j}$, where i,j are fresh for P and M, then there exists a system M' such that:*
   (a) *$O = M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$;*
   (b) *$M \xrightarrow{\tau} M'$.*
2. *For all ambients n and processes R, if $\{i,j\} \cap \mathrm{fn}(P) = \emptyset$, then*

$$n[(\boldsymbol{\nu} i,j)\mathsf{spy}_\alpha\langle i,j,P\rangle \mid R] \ \cong_{\mathrm{s}} \ n[P \mid R] \ .$$

We need a simple result that allows to garbage collect empty ambients whose name is secret.

Lemma 3.12 *$(\boldsymbol{\nu} n)n[\,] \cong_{\mathrm{s}} \mathbf{0}$.*

We also need a simple result on arbitrary contexts.

Lemma 3.13 *Let $C[-]$ and $C'[-]$ be arbitrary contexts, P and P' processes, and r a name fresh for $C[-]$ and P, such that $C[r[P]] \xrightarrow{\tau} C'[r[P']]$. Then $C[\mathbf{0}] \Rightarrow C'[\mathbf{0}]$.*

We can finally prove the correspondence between actions and contexts.

Lemma 3.14 *Let M be a system, $\alpha \in \{k.\texttt{enter\_}n, k.\texttt{exit\_}n, n.\overline{\texttt{enter}}\texttt{\_}k, k.\texttt{open\_}n\}$, and i, j fresh names for M. For all processes P with $\{i,j\} \cap \mathrm{fn}(P) = \emptyset$, if*

$$C_\alpha[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \Rightarrow\!\equiv N \mid \mathsf{done}[\,] \quad \textit{and} \quad N \Downarrow_{i,j}$$

then there exists a system M' such that $M \stackrel{\alpha}{\Longrightarrow} M'$ and $M' \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \cong_{\mathrm{s}} N$.

Proof The proof depends on the precise definition of the context. The main argument is that in the reduction

$$C_\alpha[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \Rightarrow\!\equiv N \mid \mathsf{done}[\,]$$

the fresh ambient $\mathsf{done}[\,]$ can only be unleashed if M performs the action α, possibly preceded or followed by some internal actions. The fresh barbs i, j assure that the process P does not take part in the reduction, and that the component $\mathsf{spy}_\alpha\langle i, j, P\rangle$ is found intact after the reduction. We proceed by case analysis on α. We detail here the case $\alpha = n.\overline{\texttt{enter}}\texttt{\_}k$, and we report all the other cases in Appendix B.

Case $\alpha = n.\overline{\texttt{enter}}\texttt{\_}k$. Observe that

$$\begin{array}{l} C_\alpha[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \equiv \\ \quad (\boldsymbol{\nu} a)(\boldsymbol{\nu} b)a[\texttt{in\_}n.k[\texttt{out\_}a.(\mathsf{spy}_\alpha\langle i, j, P\rangle \mid b[\texttt{out\_}k.\texttt{out\_}n.\mathsf{done}[\texttt{out\_}b]])]] \mid M \end{array}$$

To unleash the ambient done, the ambient a must use its $\texttt{in\_}n$ capability, and the ambient k must use its $\texttt{out\_}a$ capability. Moreover, the ambient b must exit from k and n, and the ambient done must exit from b. More precisely, there must exist a system M_1 and system contexts $D[-]$, $D'[-]$, and $D''[-_1, -_2, -_3]$ (for convenience we use a ternary context) such that

$$\begin{array}{ll} & C_{n.\overline{\texttt{enter}}\texttt{\_}k}[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \\ \equiv & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b)a[\texttt{in\_}n.k[\texttt{out\_}a.(\mathsf{spy}_\alpha\langle i, j, P\rangle \mid b[\texttt{out\_}k.\texttt{out\_}n.\mathsf{done}[\texttt{out\_}b]])]] \mid M \\ \Rightarrow & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b)a[\texttt{in\_}n.k[\texttt{out\_}a.(\mathsf{spy}_\alpha\langle i, j, P\rangle \mid b[\texttt{out\_}k.\texttt{out\_}n.\mathsf{done}[\texttt{out\_}b]])]] \mid M_1 \\ \stackrel{\tau}{\longrightarrow} & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b)D[a[k[\texttt{out\_}a.(\mathsf{spy}_\alpha\langle i, j, P\rangle \mid b[\texttt{out\_}k.\texttt{out\_}n.\mathsf{done}[\texttt{out\_}b]]])]] \\ \stackrel{\tau}{\Longrightarrow} & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b)D'[k[\mathsf{spy}_\alpha\langle i, j, P\rangle \mid b[\texttt{out\_}k.\texttt{out\_}n.\mathsf{done}[\texttt{out\_}b]]] \mid a[\,]] \quad (\star) \\ \Rightarrow & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b)D''[\mathsf{spy}_\alpha\langle i, j, P\rangle \mid a[\,], \mathsf{done}[\,], b[\,]] \quad (\star\star) \\ \equiv & N \mid \mathsf{done}[\,] \end{array}$$

We know that the ambient done must end up at top level (up to $\equiv$). This implies that we first consume the capability $\texttt{out\_}a$ (in the reduction sequence $(\star)$) and then the capabilities $\texttt{out\_}k$, $\texttt{out\_}n$, and $\texttt{out\_}b$ (in the reduction sequence $(\star\star)$). Moreover, as $N \Downarrow_{i,j}$, by Lemma 3.11, the process $\mathsf{spy}_\alpha\langle i, j, P\rangle$ must remain intact inside ambient k which can not be opened (although some ambients may enter k).

By examining the above reductions sequence from $C_{n.\overline{\texttt{enter}}\texttt{\_}k}[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle$ we conclude that

$$M \Rightarrow M_1 \xrightarrow{n.\overline{\texttt{enter}}\texttt{\_}k} D[k[\,\circ\,]] \Rightarrow D'[k[\,\circ\,]].$$

As names a, b, and done are all fresh, by Lemma 3.13 there is M' such that:

$$D'[k[\,\circ\,]] \Rightarrow M' \equiv D''[\circ \mid \mathbf{0}, \mathbf{0}, \mathbf{0}].$$

As a and b are fresh and, by Lemma 3.12 $(\boldsymbol{\nu} n)n[\,] \cong_{\mathrm{s}} \mathbf{0}$, it holds that

$$M' \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \equiv D''[\mathsf{spy}_\alpha\langle i, j, P\rangle \mid \mathbf{0}, \mathbf{0}, \mathbf{0}] \cong_{\mathrm{s}} (\boldsymbol{\nu} a)(\boldsymbol{\nu} b)D''[\mathsf{spy}_\alpha\langle i, j, P\rangle \mid a[\,], \mathbf{0}, b[\,]]$$

and hence also that:

$$\begin{array}{rcl} M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \mid \mathsf{done}[\,] & \cong_{\mathrm{s}} & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b) D''[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid a[\,], \mathbf{0}, b[\,]] \mid \mathsf{done}[\,] \\ & \equiv & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b) D''[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid a[\,], \mathsf{done}[\,], b[\,]] \\ & \equiv & N \mid \mathsf{done}[\,] \end{array}$$

As done is a fresh name and $\cong_{\mathrm{s}}$ is closed under restriction we have $M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \cong_{\mathrm{s}} N$, as desired. □

When proving the completeness result we implicitly use a standard property of reduction barbed congruence.

Proposition 3.15 *If $P \cong_{\mathrm{s}} Q$ then*

- *$P \Downarrow n$ iff $Q \Downarrow n$*
- *$P \Rightarrow P'$ implies there is Q' such that $Q \Rightarrow Q'$ and $P' \cong_{\mathrm{s}} Q'$.*

Similar results hold for reduction barbed congruence over processes. In the sequel we will use these properties without comment.

Theorem 3.16 (Completeness) *Reduction barbed congruence over systems is contained in late bisimilarity.*

Proof We prove that the relation $\mathcal{R} = \{(M,N) \mid M \cong_{\mathrm{s}} N\}$ is a late bisimulation. The result will then follow by co-induction.

- Suppose $M \;\mathcal{R}\; N$. Suppose also that $M \xrightarrow{\alpha} M'$, with $\alpha \in \{k.\mathtt{enter}\_n, k.\mathtt{exit}\_n, n.\overline{\mathtt{enter}}\_k, k.\mathtt{open}\_n\}$. We must find a system N' such that $N \stackrel{\alpha}{\Longrightarrow} N'$ and for all P, $M' \bullet P \cong_{\mathrm{s}} N' \bullet P$.

  The idea of the proof is to use a particular context which mimics the effect of the action α, and also allows us to subsequently compare the residuals of the two systems. This context has the form

  $$D_\alpha\langle P\rangle[-] \;=\; (C_\alpha[-] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \mid \mathsf{Flip}$$

  where $C_\alpha[-]$ are the contexts in Table 7 and Flip is the system:

  $$(\boldsymbol{\nu} k) k[\mathtt{in\_done}.\mathtt{out\_done}.(\mathsf{succ}[\mathtt{out}\_k] \oplus \mathsf{fail}[\mathtt{out}\_k])]$$

  where succ and fail are fresh names. Intuitively, the existence of the fresh barb fail indicates that the action α has not yet happened, whereas the presence of succ together with the absence of fail ensures that the action α has been performed, and has been reported via done.

  As $\cong_{\mathrm{s}}$ is preserved by system contexts, $M \cong_{\mathrm{s}} N$ implies that, for all processes P, it holds

  $$D_\alpha\langle P\rangle[M] \;\cong_{\mathrm{s}}\; D_\alpha\langle P\rangle[N]\ .$$

  By Lemma 3.9 and 3.11(1), we can build the following reduction sequence:

  $$D_\alpha\langle P\rangle[M] = (C_\alpha[M] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \mid \mathsf{Flip} \;\Rightarrow M_1 \mid \mathsf{Flip} \;\Rightarrow O_1$$

with $M_1 \equiv D'[\mathsf{spy}_\alpha\langle i,j,P\rangle] \mid \mathsf{done}[\,] \cong_{\mathrm{s}} (M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \mid \mathsf{done}[\,]$, for some system context $D'[-]$, and by Lemma 3.10 $O_1 \cong_{\mathrm{s}} (M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \mid \mathsf{done}[\,] \mid \mathsf{succ}[\,]$ with $O_1 \Downarrow_{i,j,\mathsf{succ}} \not\Downarrow_{\mathsf{fail}}$.

This reduction must be matched by a corresponding reduction sequence

$$D_\alpha\langle P\rangle[N] \;\Rightarrow\; O_2$$

where $O_1 \cong_{\mathrm{s}} O_2$ and hence $O_2 \Downarrow_{i,j,\mathsf{succ}} \not\Downarrow_{\mathsf{fail}}$.

The constrains on the barbs allow us to deduce the structure of the above reduction sequence. That is:

$$D_\alpha\langle P\rangle[N] = (C_\alpha[N] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \mid \mathsf{Flip} \;\Rightarrow N_1 \mid \mathsf{Flip} \;\Rightarrow O_2$$

with $N_1 \equiv D''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \mid \mathsf{done}[\,]$, and $O_2 \cong_{\mathrm{s}} D'''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \mid \mathsf{done}[\,] \mid \mathsf{succ}[\,]$ for some system contexts $D''[-]$ and $D'''[-]$ with $D''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \;\Rightarrow D'''[\mathsf{spy}_\alpha\langle i,j,P\rangle]$.

As $C_\alpha[N] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \;\Rightarrow N_1 \equiv D''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \mid \mathsf{done}[\,]$, by Lemma 3.14 there is N' such that $N \stackrel{\alpha}{\Longrightarrow} N'$ and $D''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \cong_{\mathrm{s}} N' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$. As $D''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \;\Rightarrow D'''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \Downarrow_{i,j}$ there is N'' such that $N' \;\Rightarrow N''$ and $D'''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \cong_{\mathrm{s}} N'' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$. Summarising, there is N'' such that $N \stackrel{\alpha}{\Longrightarrow} N''$ and:

- $O_1 \cong_{\mathrm{s}} M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \mid \mathsf{done}[\,] \mid \mathsf{succ}[\,]$
- $O_2 \cong_{\mathrm{s}} D'''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \mid \mathsf{done}[\,] \mid \mathsf{succ}[\,]$
- $D'''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \cong_{\mathrm{s}} N'' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$
- $O_1 \cong_{\mathrm{s}} O_2$.

As barbed congruence is preserved by restriction, we have

$$(\boldsymbol{\nu}\mathsf{done},\mathsf{succ})O_1 \cong_{\mathrm{s}} (\boldsymbol{\nu}\mathsf{done},\mathsf{succ})O_2 \;.$$

By Lemma 3.12 $(\boldsymbol{\nu}\mathsf{done})\mathsf{done}[\,] \cong_{\mathrm{s}} (\boldsymbol{\nu}\mathsf{succ})\mathsf{succ}[\,] \cong_{\mathrm{s}} \mathbf{0}$, which implies

$$M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \;\cong_{\mathrm{s}}\; N'' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle.$$

Again, $\cong_{\mathrm{s}}$ is preserved by restriction and, by Lemma 3.11(2), we can finally derive $M' \bullet P \;\mathcal{R}\; N'' \bullet P$, for all processes P.

- Suppose now $M \;\mathcal{R}\; N$ and $M \xrightarrow{*.\mathtt{enter}\_n} M'$, We must find a system N' such that $N \mid n[\circ] \;\Rightarrow N'$ and for all P, $M' \bullet P \cong_{\mathrm{s}} N' \bullet P$.

  We consider the context

$$C\langle P\rangle[-] \;=\; - \mid n[\mathsf{spy}_{*.\mathtt{enter}\_n}\langle i,j,P\rangle] \;.$$

  Because $\cong_{\mathrm{s}}$ is preserved by system contexts, for all processes P it holds

$$C\langle P\rangle[M] \;\cong_{\mathrm{s}}\; C\langle P\rangle[N] \;.$$

  By inspecting the reduction rules of $C\langle P\rangle[M]$ we observe that,

$$C\langle P\rangle[M] \;\Rightarrow\; M' \bullet \mathsf{spy}_{*.\mathtt{enter}\_n}\langle i,j,P\rangle$$

where $M' \bullet \mathsf{spy}_{*.\mathtt{enter}\_n}\langle i, j, P\rangle \Downarrow_{i,j}$. Call this outcome O_1.

This reduction must be matched by a corresponding reduction

$$C\langle P\rangle[N] \;\Rightarrow\; O_2$$

where $O_1 \cong_{\mathrm{s}} O_2$ and $O_2 \Downarrow_{i,j}$. By Lemma 3.11(1) it follows that there is a system N' such that $O_2 = N' \bullet \mathsf{spy}_{*.\mathtt{enter}\_n}\langle i, j, P\rangle$ and $N \mid n[\,\circ\,] \;\Rightarrow N'$. Again, as $\cong_{\mathrm{s}}$ is preserved by restriction, from $O_1 \cong_{\mathrm{s}} O_2$ and Lemma 3.11(2) we can derive $M' \bullet P \cong_{\mathrm{s}} N' \bullet P$, for all P, as required.

- Suppose $M \;\mathcal{R}\; N$ and $M \xrightarrow{*.\mathtt{exit}\_n} M'$. In this case we must find a system N' such that $n[\circ \mid N] \;\Rightarrow N'$ and for all P, $M' \bullet P \cong_{\mathrm{s}} N' \bullet P$.

  We consider the context

  $$C\langle P\rangle[-] \;=\; n[- \mid \mathsf{spy}_{*.\mathtt{exit}\_n}\langle i, j, P\rangle] \;.$$

  Because $\cong_{\mathrm{s}}$ is preserved by system contexts, for all processes P it holds

  $$C\langle P\rangle[M] \;\cong_{\mathrm{s}}\; C\langle P\rangle[N] \;.$$

  By inspecting the reduction rules of $C\langle P\rangle[M]$ we observe that,

  $$C\langle P\rangle[M] \;\Rightarrow\; M' \bullet \mathsf{spy}_{*.\mathtt{exit}\_n}\langle i, j, P\rangle$$

  where $M' \bullet \mathsf{spy}_{*.\mathtt{exit}\_n}\langle i, j, P\rangle \Downarrow_{i,j}$. Call this outcome O_1.

  This reduction must be matched by a corresponding reduction

  $$C\langle P\rangle[N] \;\Rightarrow\; O_2$$

  where $O_1 \cong_{\mathrm{s}} O_2$ and $O_2 \Downarrow_{i,j}$. By Lemma 3.11(1) it follows that there is a system N' such that $O_2 = N' \bullet \mathsf{spy}_{*.\mathtt{enter}\_n}\langle i, j, P\rangle$ and $n[\circ \mid N] \;\Rightarrow N'$. Again, as $\cong_{\mathrm{s}}$ is preserved by restriction, from $O_1 \cong_{\mathrm{s}} O_2$ and Lemma 3.11(2) we can derive $M' \bullet P \cong_{\mathrm{s}} N' \bullet P$, for all P, as required.

This concludes the analysis. □

As a consequence:

Theorem 3.17 (Characterisation of $\cong_{\mathrm{s}}$) *Late bisimilarity, early bisimilarity, and reduction barbed congruence over systems coincide.*

Proof Theorem 3.8 states that $\approx \subseteq \approx_e$ and $\approx_e \subseteq \cong_{\mathrm{s}}$. Theorem 3.16 states the reduction barbed congruence over systems is contained in late bisimilarity, that is $\cong_{\mathrm{s}} \subseteq \approx$. We hence have the following chain of inclusions $\cong_{\mathrm{s}} \subseteq \approx \subseteq \approx_e \subseteq \cong_{\mathrm{s}}$. □

A remark on transitivity of (late) bisimilarity. Giving a direct proof that $\approx$ is a transitive relation seems to be awkward. At the same time, the characterisation result does not rely on the transitivity of $\approx$. As $\cong_{\mathrm{s}}$ is trivially an equivalence relation, late and early bisimilarity are also equivalence relations.

4 Up-to Proof Techniques

In the previous section we presented a labelled characterisation of reduction barbed congruence to prove that two systems have the same behaviour. In this section we adapt some well-known *up-to* proof techniques [28, 33] to our setting. These techniques allow us to reduce the size of the relation $\mathcal{R}$ to exhibit to prove that two processes are bisimilar. We focus on two forms of up-to techniques: the *up-to expansion* [35] and the *up-to context* technique [32]. As in the π-calculus, these two techniques can be merged.

The expansion [2], written $\lesssim$, is an asymmetric variant of the bisimilarity that allows us to count the number of silent moves performed by a system. Intuitively, $M \lesssim N$ holds if M and N are bisimilar and N has at least as many τ-moves as M. Formally,

Definition 4.1 (Expansion) *A relation $\mathcal{R}$ over systems is an* expansion *if $M\ \mathcal{R}\ N$ implies:*

- *if $M \xrightarrow{\alpha} M'$, $\alpha \notin \{*.\mathtt{enter}\_n, *.\mathtt{exit}\_n\}$, then there exists a system N' such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} N'$ and for all processes P it holds $M' \bullet P\ \mathcal{R}\ N' \bullet P$;*
- *if $M \xrightarrow{*.\mathtt{enter}\_n} M'$ then there exists a system N' such that $N \mid n[\,\circ\,] \Rightarrow N'$ and for all processes P it holds $M' \bullet P\ \mathcal{R}\ N' \bullet P$;*
- *if $M \xrightarrow{*.\mathtt{exit}\_n} M'$ then there exists a system N' such that $n[\circ \mid N] \Rightarrow N'$ and for all processes P it holds $M' \bullet P\ \mathcal{R}\ N' \bullet P$;*
- *if $N \xrightarrow{\alpha} N'$, $\alpha \notin \{*.\mathtt{enter}\_n, *.\mathtt{exit}\_n\}$, then there exists a system M' such that $M \xrightarrow{\hat{\alpha}} M'$ and for all processes P it holds $M' \bullet P\ \mathcal{R}\ N' \bullet P$;*
- *if $N \xrightarrow{*.\mathtt{enter}\_n} N'$ then $(M \mid n[P])\ \mathcal{R}\ N' \bullet P$, for all processes P;*
- *if $N \xrightarrow{*.\mathtt{exit}\_n} N'$ then $n[M \mid P]\ \mathcal{R}\ N' \bullet P$, for all processes P.*

We write $M \lesssim N$, if $M\ \mathcal{R}\ N$ for some expansion $\mathcal{R}$.

Definition 4.2 (Bisimulation up to context and up to $\gtrsim \approx$) *A symmetric relation $\mathcal{R}$ over systems is a* bisimulation up to context and up to $\gtrsim \approx$ *if $M\ \mathcal{R}\ N$ implies:*

- *if $M \xrightarrow{\alpha} M''$, $\alpha \notin \{*.\mathtt{enter}\_n, *.\mathtt{exit}\_n\}$, then there exists a system N'' such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} N''$, and for all processes P there is a system context $C[-]$ and systems M' and N' such that $M'' \bullet P \gtrsim C[M']$, $N'' \bullet P \approx C[N']$, and $M'\ \mathcal{R}\ N'$;*
- *if $M \xrightarrow{*.\mathtt{enter}\_n} M''$ then there exists a system N'' such that $N \mid n[\,\circ\,] \Rightarrow N''$, and for all processes P there is a system context $C[-]$ and systems M' and N' such that $M'' \bullet P \gtrsim C[M']$, $N'' \bullet P \approx C[N']$, and $M'\ \mathcal{R}\ N'$;*

- *if* $M \xrightarrow{*.\mathtt{exit}\_n} M''$ *then there exist a system* N'' *such that* $n[\circ \mid N] \Rightarrow N''$*, and for all processes* P *there is a system context* $C[-]$ *and systems* M' *and* N' *such that* $M'' \bullet P \gtrsim C[M']$*,* $N'' \bullet P \approx C[N']$*, and* $M' \mathcal{R} N'$*.*

Theorem 4.3 *If* $\mathcal{R}$ *is a bisimulation up to context and up to* $\gtrsim \approx$*, then* $\mathcal{R} \subseteq \approx$*.*

Proof We define the relation $\mathcal{S}$ as the smallest relation such that:

1. $M \mathcal{R} N$ implies $M \mathcal{S} N$;
2. $M \gtrsim A$, $A \mathcal{S} B$, $B \approx N$ implies $M \mathcal{S} N$;
3. $M \mathcal{S} N$ implies $C[M] \mathcal{S} C[N]$, for all system contexts $C[-]$.

We prove that $\mathcal{S}$ is a late bisimulation, by induction on its definition. This will assure the soundness of the relation $\mathcal{R}$, because $M \mathcal{R} N$ implies $M \mathcal{S} N$ which implies $M \approx N$. Observe that $\mathcal{S}$ is symmetric because $\mathcal{R}$ is.

- $M \mathcal{S} N$ because $M \mathcal{R} N$.

  Suppose that $M \xrightarrow{\alpha} M''$, with $\alpha \notin \{*.\mathtt{enter}\_n, *.\mathtt{exit}\_n\}$. As $\mathcal{R}$ is a bisimulation up to context and up-to $\gtrsim$, we know that there exists a system N'' such that $N \stackrel{\alpha}{\Longrightarrow} N''$. We also know that for all process P, there exist a system context $C[-]$ and systems M' and N' such that $M'' \bullet P \gtrsim C[M']$, $N'' \bullet P \gtrsim C[N']$, and $M' \mathcal{R} N'$. This implies $M' \mathcal{S} N'$. By construction $\mathcal{S}$ is preserved by system contexts and $C[M'] \mathcal{S} C[N']$ holds. By construction $\mathcal{S}$ is closed under expansion, and therefore $M'' \mathcal{S} N''$, as required.

  Suppose that $M \xrightarrow{*.\mathtt{enter}\_n} M''$. As $\mathcal{R}$ is a bisimulation up to context and up to $\gtrsim$, we know that there exists a system N'' such that $N \mid n[\circ] \stackrel{*.\mathtt{enter}\_n}{\Longrightarrow} N''$. We also know that for all process P, there exist a system context $C[-]$ and systems M' and N' such that $M'' \bullet P \gtrsim C[M']$, $N'' \bullet P \gtrsim C[N']$, and $M' \mathcal{R} N'$. This implies $M' \mathcal{S} N'$. By construction, $\mathcal{S}$ is preserved by system contexts, and $C[M'] \mathcal{S} C[N']$ holds. By construction $\mathcal{S}$ is closed under expansion, and therefore $M'' \mathcal{S} N''$, as required.

  Suppose that $M \xrightarrow{*.\mathtt{exit}\_n} M''$. As $\mathcal{R}$ is a bisimulation up to context and up to $\gtrsim$, we know that there exists a system N'' such that $n[\circ \mid N] \stackrel{*.\mathtt{exit}\_n}{\Longrightarrow} N''$. We also know that for all process P, there exist a system context $C[-]$ and systems M' and N' such that $M'' \bullet P \gtrsim C[M']$, $N'' \bullet P \gtrsim C[N']$, and $M' \mathcal{R} N'$. This implies $M' \mathcal{S} N'$. By construction, $\mathcal{S}$ is preserved by system contexts, and $C[M'] \mathcal{S} C[N']$ holds. By construction $\mathcal{S}$ is closed under expansion, and we conclude $M'' \mathcal{S} N''$, as required.

- $M \mathcal{S} N$ because $M \gtrsim A, A \mathcal{S} B, B \approx N$.

  The induction hypothesis tells us that $A \mathcal{S} B$ behaves like a late bisimulation.

  Suppose $M \xrightarrow{\alpha} M'$, with $\alpha \notin \{*.\mathtt{enter}\_n, *.\mathtt{exit}\_n\}$. A simple diagram chasing allows us to conclude that there are systems A', B', N' such that for all process P it holds $M' \bullet P \gtrsim A' \bullet P \mathcal{S} B' \bullet P \approx N' \bullet P$, and in turn, by construction of $\mathcal{S}$, $M' \bullet P \mathcal{S} N' \bullet P$.

  Suppose $M \xrightarrow{*.\mathtt{enter}\_n} M'$. As $M \gtrsim A$, for all process P, it holds $M' \bullet P \gtrsim A \mid n[P]$. As $A \mathcal{S} B$, the closure properties of $\mathcal{S}$ assure that $A \mid n[P] \mathcal{S} B \mid n[P]$. Late bisimilarity is preserved by system contexts, and, since $B \approx N$, we conclude that $B \mid n[P] \approx N \mid n[P]$.

But $N \mid n[P] \;\Rightarrow N \mid n[P]$, and $M' \bullet P \gtrsim \mathcal{S} \approx (N \mid n[\,\circ\,]) \bullet P$. This, by construction of $\mathcal{S}$, implies $M' \bullet P \;\mathcal{S}\; (N \mid n[\,\circ\,]) \bullet P$.

Suppose $M \xrightarrow{*.\mathtt{exit}\_n} M'$. As $M \gtrsim A$, for all process P, it holds $M' \bullet P \gtrsim n[P \mid A]$. As $A \;\mathcal{S}\; B$, the closure properties of $\mathcal{S}$ assure that $n[P \mid A] \;\mathcal{S}\; n[P \mid B]$. Late bisimilarity is preserved by system contexts, and since $B \approx N$ we conclude that $n[P \mid A] \approx n[P \mid N]$. But $n[P \mid B] \;\Rightarrow n[P \mid N]$, and $M' \bullet P \gtrsim \mathcal{S} \approx n[\circ \mid N] \bullet P$. This, by construction of $\mathcal{S}$, implies $M' \bullet P \;\mathcal{S}\; n[\circ \mid N] \bullet P$.

- $C[M] \;\mathcal{S}\; C[N]$ because $M \;\mathcal{S}\; N$ and $C[-]$ is a system context.

  The induction hypothesis tells us that $(M, N) \in \mathcal{S}$ is a pair satisfying the bisimulation conditions in $\mathcal{S}$. Lemma 3.4 assures that the pair $(C[M], C[N]) \in \mathcal{S}$ satisfies the bisimulation conditions in $\mathcal{S}$.

This completes the induction. □

5 A Semantic Theory for Processes

In this section we characterise reduction barbed congruence over processes, $\cong_{\mathrm{p}}$, in terms of our labelled bisimilarity over systems, $\approx$.

The relation $\cong_{\mathrm{p}}$ is closed under arbitrary process contexts: reducing the number of contexts in the quantification is a first step towards the definition of a useful proof technique, and, broadly speaking, towards an understanding of the behavioural theory of processes.

We show that it is possible to work with a lighter definition of contextuality. In particular it suffices to require closure under the two crucial operators of MA: parallel composition (to model concurrency) and ambient construct (to model locality).

Definition 5.1 Reduction barbed equivalence over processes*, written* $\cong_{\mathrm{p}}^{\mathrm{e}}$*, is the largest symmetric relation over processes which is reduction closed, barb preserving, and closed under parallel composition and ambient construct.*

Theorem 5.2 (Context Lemma) *The relations* $\cong_{\mathrm{p}}$ *and* $\cong_{\mathrm{p}}^{\mathrm{e}}$ *coincide.*

Reduction barbed equivalence over processes still requires a universal quantification on non-trivial contexts. More than that, a direct proof of the above context lemma is surprisingly difficult. We look for a more operative characterisation of $\cong_{\mathrm{p}}^{\mathrm{e}}$, and we postpone the proof of the context lemma after Theorem 5.3.

Theorem 5.3 (Characterisation of $\cong_{\mathrm{p}}^{\mathrm{e}}$**)** *Let*

$$\mathcal{S} = \{(P, Q) :\; k[P \mid R] \approx k[Q \mid R],\; \textit{for all } k, R\}\ .$$

The relations $\cong_{\mathrm{p}}^{\mathrm{e}}$ *and* $\mathcal{S}$ *coincide.*

To prove Theorem 5.3 we need some technical lemmas. The next two lemmas (their proofs are reported in the Appendix C) are necessary for proving the completeness part of Theorem 5.3. In particular Lemma 5.4 says that reduction barbed equivalence over processes is preserved by restriction. This result will be also useful when proving the context lemma.

Lemma 5.4 *If $P \cong^{\mathrm{e}}_{\mathrm{p}} Q$, then $(\boldsymbol{\nu} n)P \cong^{\mathrm{e}}_{\mathrm{p}} (\boldsymbol{\nu} n)Q$.*

Lemma 5.5 *$\cong^{\mathrm{e}}_{\mathrm{p}} \cap (\mathcal{M} \times \mathcal{M}) \subseteq \cong_{\mathrm{s}}$, where $\mathcal{M}$ is the set of all systems.*

The following lemma is similar to Lemma 3.11(2) where we use a slight simplification of the spy-contexts given in Table 8.

Lemma 5.6 *Let $\mathsf{spy}\langle i, j, - \rangle \stackrel{\text{def}}{=} (i[] \mid -) \oplus (j[] \mid -)$, for i and j fresh, then:*

$$n[P \mid R] \approx n[(\boldsymbol{\nu} i, j)\mathsf{spy}\langle i, j, P\rangle \mid R] \ .$$

Proof Similar to the proof of Lemma 3.11(2). □

The following technical lemma is crucial in the proof of Theorem 5.3.

Lemma 5.7 *Let P and Q be two processes, and k an ambient name, with $k \notin \mathrm{fn}(P, Q)$. If $k[P] \approx k[Q]$, then for all n, R it holds $n[P \mid R] \approx n[Q \mid R]$.*

Proof By two applications of Lemma 5.6 we derive $k[\mathsf{spy}\langle i, j, P\rangle] \approx k[\mathsf{spy}\langle i, j, Q\rangle]$ from $k[P] \approx k[Q]$. Then, the definition of bisimulation assures us that if $k[P] \xrightarrow{k.\mathtt{open}\_n} n[\mathsf{spy}\langle i, j, P\rangle \mid \circ]$ then there is a matching transition $k[Q] \xrightarrow{k.\mathtt{open}\_n} n[\mathsf{spy}\langle i, j, Q\rangle \mid \circ]$, and that for all R it holds $n[\mathsf{spy}\langle i, j, P\rangle \mid R] \approx n[\mathsf{spy}\langle i, j, Q\rangle \mid R]$. Remark that the *spy* context ensures that the matching transition must be a strong transition, otherwise one of the two barbs i or j would be lost in the outcome. Up to alpha-conversion we assume $\{i, j\} \cap R = \emptyset$, for all R. As bisimulation is closed by restriction and under structural congruence, we have $n[(\boldsymbol{\nu} i, j)\mathsf{spy}\langle i, j, P\rangle \mid R] \approx n[(\boldsymbol{\nu} i, j)\mathsf{spy}\langle i, j, Q\rangle \mid R]$. By Lemma 5.6 we obtain $n[P \mid R] \approx n[Q \mid R]$, as required. □

Everything is now in place to prove Theorem 5.3.

Proof of Theorem 5.3. We first show that $P \cong^{\mathrm{e}}_{\mathrm{p}} Q$ implies $P \mathcal{S} Q$. For that, we must show that for all k, R, it holds $k[P \mid R] \approx k[Q \mid R]$. Both $k[P \mid R]$ and $k[Q \mid R]$ are systems, and it holds $k[P \mid R] \cong^{\mathrm{e}}_{\mathrm{p}} k[Q \mid R]$ because $\cong^{\mathrm{e}}_{\mathrm{p}}$ is closed under parallel composition and ambient construct. The result follows from Lemma 5.5 and Theorem 3.17.

It remains to prove that $\mathcal{S} \subseteq \cong^{\mathrm{e}}_{\mathrm{p}}$. For that, we must show that $\mathcal{S}$ is reduction closed, barb preserving, and closed under parallel composition and ambient construct.

1. *$\mathcal{S}$ is reduction closed.* Suppose $P \mathcal{S} Q$ and $P \rightarrowtriangle P'$. Let n be a name such that $n \notin \mathrm{fn}(P, Q)$.

   We have $n[P] \approx n[Q]$, by definition of $\mathcal{S}$. As $n \notin \mathrm{fn}(P, Q)$, and because of the correspondence between τ-transitions and reductions, there is a system M such that $n[P] \xrightarrow{\tau} M \equiv n[P']$. As $n[P] \approx n[Q]$, there is N such that $n[Q] \Rightarrow N$ and $M \approx N$. As $n \notin \mathrm{fn}(P, Q)$, there must be Q' such that $Q \rightarrowtriangle^* Q'$ and $N \equiv n[Q']$; thus $n[P'] \approx n[Q']$. Lemma 5.7 allows us to derive $P' \mathcal{S} Q'$, as desired.

2. *$\mathcal{S}$ is barb preserving.* Suppose that $P \mathcal{S} Q$ and $P \Downarrow n$. Consider the context

   $$C[-] = b[- \mid a[\mathtt{in}\_n.\mathtt{out}\_n.\mathtt{ok}[\mathtt{out}\_a.\mathtt{out}\_b]]]$$

   where a, b and ok are fresh for both P and Q. Then $C[P] \approx C[Q]$ by definition of $\mathcal{S}$. As $P \Downarrow n$, the construction of $C[-]$ assures that $C[P] \Downarrow \mathsf{ok}$. Bisimilarity is barb preserving and $C[Q] \Downarrow \mathsf{ok}$ must hold. The construction of $C[-]$ guarantees that $Q \Downarrow n$.

3. *$\mathcal{S}$ is closed under parallel composition and ambient construct.*

- $P \mathcal{S} Q$ implies $P \mid R \mathcal{S} Q \mid R$.
  By definition of $\mathcal{S}$ we have $k[P \mid R'] \approx k[Q \mid R']$ for all k, R'. By taking $R' = R \mid R''$ for arbitrary R'' we have $k[P \mid R \mid R''] \approx k[Q \mid R \mid R'']$ for all R''. This implies $P \mid R \mathcal{S} Q \mid R$.
- $P \mathcal{S} Q$ implies $n[P] \mathcal{S} n[Q]$.
  By definition of $\mathcal{S}$ we have $n[P] \approx n[Q]$ for all n. The result follows from the closure of $\approx$ under static contexts. □

The characterisation of $\cong^{\mathrm{e}}_{\mathrm{p}}$ is a fundamental tool to reason about processes. As a first application, we prove the context lemma.

Proof of Theorem 5.2. We have to show that $\cong^{\mathrm{e}}_{\mathrm{p}} = \cong_{\mathrm{p}}$. The inclusion $\cong_{\mathrm{p}} \subseteq \cong^{\mathrm{e}}_{\mathrm{p}}$ is straightforward. For the converse we must prove that

1. $\cong^{\mathrm{e}}_{\mathrm{p}}$ is reduction closed;
2. $\cong^{\mathrm{e}}_{\mathrm{p}}$ is barb preserving;
3. $\cong^{\mathrm{e}}_{\mathrm{p}}$ is closed under arbitrary contexts.

Conditions 1 and 2 hold by definition of $\cong^{\mathrm{e}}_{\mathrm{p}}$. It remains to show that the relation $\cong^{\mathrm{e}}_{\mathrm{p}}$ is preserved by all process contexts. The relation $\cong^{\mathrm{e}}_{\mathrm{p}}$ is preserved by parallel composition and ambient constructor by definition. It is also preserved by restriction by Lemma 5.4. It remains to prove that it is preserved by prefixing and replicated prefixing. We report the proof that $\cong^{\mathrm{e}}_{\mathrm{p}}$ is preserved by prefixing in the Appendix, and we focus on replicated prefixing.

We have to prove that if $P \cong^{\mathrm{e}}_{\mathrm{p}} Q$, then $!\pi.P \cong^{\mathrm{e}}_{\mathrm{p}} !\pi.Q$. Rather than working directly with $\cong^{\mathrm{e}}_{\mathrm{p}}$, we use Theorem 5.3 and we prove that $!\pi.P \mathcal{S} !\pi.Q$. For that, we show that $k[!\pi.P \mid R] \approx k[!\pi.Q \mid R]$ for all k and R. We perform a case analysis on π.

Suppose that $\pi = \mathtt{in}\_o$. We show that the relation

$$\mathcal{R} = \{(n[!\mathtt{in}\_o.P \mid R], n[!\mathtt{in}\_o.Q \mid R]) \ : \ P \cong^{\mathrm{e}}_{\mathrm{p}} Q\}^{=} \cup \ \approx$$

is a *bisimulation up to context and up to* $\gtrsim \approx$.

The most interesting case is when the process $!\mathtt{in}\_o.P$ exercises the capability $\mathtt{in}\_o$. Suppose

$$n[!\mathtt{in}\_o.P \mid R] \xrightarrow{n.\mathtt{enter}\_o} o[n[P \mid !\mathtt{in}\_o.P \mid R] \mid \circ] \ .$$

We have a matching transition

$$n[!\mathtt{in}\_o.Q \mid R] \xrightarrow{n.\mathtt{enter}\_o} o[n[Q \mid !\mathtt{in}\_o.Q \mid R] \mid \circ] \ .$$

Since $P \cong^{\mathrm{e}}_{\mathrm{p}} Q$, we have $P \mathcal{S} Q$ and in turn, for all R', we have $n[P \mid R'] \approx n[Q \mid R']$. As $\approx$ is preserved by system contexts, for all instantiations of $\circ$ it holds $o[n[P \mid R'] \mid \circ] \approx o[n[Q \mid R'] \mid \circ]$. By taking $R' = !\mathtt{in}\_o.Q \mid R$, we obtain

$$o[n[!\mathtt{in}\_o.Q \mid R \mid P] \mid \circ] \approx o[n[Q \mid !\mathtt{in}\_o.Q \mid R] \mid \circ] \ .$$

Then, for all processes S, the following hold:

$$\begin{array}{lll} o[n[P \mid !\mathtt{in}\_o.P \mid R] \mid \circ] \bullet S & \gtrsim & C[n[!\mathtt{in}\_o.P \mid R \mid P]] \\ o[n[Q \mid !\mathtt{in}\_o.Q \mid R] \mid \circ] \bullet S & \approx & C[n[!\mathtt{in}\_o.Q \mid R \mid P]] \end{array}$$

where $C[-] = o[- \mid S]$ (we can rearrange the terms using structural congruence because $\equiv \subseteq \gtrsim$ and $\equiv \subseteq \approx$). By construction of $\mathcal{R}$ we have

$$n[!\texttt{in\_}o.P \mid R \mid P] \;\mathcal{R}\; n[!\texttt{in\_}o.Q \mid R \mid P]$$

and we can conclude that up to context and up to $\gtrsim \approx$ we are still in $\mathcal{R}$.

The cases $\pi = \texttt{out\_}o$ and $\pi = \texttt{open\_}o$ follow along similar lines. □

The result below is a consequence of Theorems 5.2 and 5.3.

Theorem 5.8 (Characterisation of $\cong_{\mathrm{p}}$) *The relations $\mathcal{S}$ and $\cong_{\mathrm{p}}$ coincide.*

The relation $\mathcal{S}$ still involves a universal quantification over all the processes R. Yet, it is built on top of $\approx$ and it can be coupled with the up-to proof techniques. In turn, it reveals a useful tool to reason about processes, as illustrated by the proof of the context lemma and by the other examples given in Section 7.

Systems revisited when working with systems, In Section 3, we conjectured that reduction barbed congruence over systems ($\cong_{\mathrm{s}}$) *is* "the right" equality when working with systems. We are now in measure to close the conjecture. In fact, if we restrict our attention to systems, we can show that system contexts have the same discriminating power as arbitrary contexts.

Theorem 5.9 *Let M and N be two systems, then $M \cong_{\mathrm{s}} N$ if and only if $M \cong_{\mathrm{p}} N$.*

Proof By definition, $M \cong_{\mathrm{p}} N$ implies $M \cong_{\mathrm{s}} N$. For the converse, by Theorem 3.17, if $M \cong_{\mathrm{s}} N$ then $M \approx N$. As $\approx$ is preserved by system contexts, for all n and R $n[M \mid R] \approx n[N \mid R]$. By Theorems 5.3 and 5.2 it follows that $M \cong_{\mathrm{p}} N$. □

This in turn implies a strong result: $\approx$ *completely characterises* $\cong_{\mathrm{p}}$ *on systems.*

Theorem 5.10 *Let M and N be two systems, then $M \cong_{\mathrm{p}} N$ if and only if $M \approx N$.*

6 Adding Communication

In this section we adapt our characterisation results to the calculus with communication. The basic idea is to have an *output process* $\langle E \rangle$, which outputs the message E, and an input process $(x).Q$ where x is bound in the continuation Q. Messages are sequences of capabilities. Unlike [6, 21] we do not allow ambient names to be transmitted. This has been a deliberate choice as, a priori, when the name is transmitted the recipient gets considerable control over that ambient.

The syntax of the extended language is given in Table 9. We assume an understanding of free and bound variables ($\mathrm{fv}(\cdot)$ and $\mathrm{bv}(\cdot)$), and of *substitutions*. A process P is said to be *closed* if $\mathrm{fv}(P) = \emptyset$; otherwise is said to be *open*. The structural and reduction rules below define the semantics of communication:

$$E.(F.P) \;\equiv\; (E.F).P \qquad \varepsilon.P \;\rightarrowtriangle\; P \qquad (x).P \mid \langle E \rangle \;\rightarrowtriangle\; P\{^{E}/_{x}\}$$

The LTS is extended by the introduction of two new pre-actions (E) for input, $\langle - \rangle$ for output, and a new form of concretion $(\boldsymbol{\nu}\tilde{m})\langle E \rangle Q$; intuitively the message E is bufferised in the concretion, Q is the outcome of the output action, and $\tilde{m}$ are the names shared by E and

Table 9 Message-passing Mobile Ambients

Names: $a, b, \ldots, k, l, m, n, \ldots \in \mathbf{N}$

Capabilities:

C	::=	$\texttt{in\_}n$	may enter into n
	$\mid$	$\texttt{out\_}n$	may exit out of n
	$\mid$	$\texttt{open\_}n$	may open n

Expressions:

E, F	::=	x	variable
	$\mid$	C	capability
	$\mid$	$E.F$	path
	$\mid$	ε	empty path

Guards:

G	::=	E	expression
	$\mid$	(x)	input

Systems:

M, N	::=	$\mathbf{0}$	termination
	$\mid$	$M_1 \mid M_2$	parallel composition
	$\mid$	$(\boldsymbol{\nu} n)M$	restriction
	$\mid$	$n[P]$	ambient

Processes:

P, Q, R	::=	$\mathbf{0}$	nil process
	$\mid$	$P_1 \mid P_2$	parallel composition
	$\mid$	$(\boldsymbol{\nu} n)P$	restriction
	$\mid$	$G.P$	prefixing
	$\mid$	$n[P]$	ambient
	$\mid$	$!G.P$	replication
	$\mid$	$\langle E \rangle$	output

Table 10 Pre-actions, Concretions and Labelled Transition System for Communication

Pre-actions: $\pi ::= \ldots \mid (E) \mid \langle - \rangle$

Concretions: $K ::= (\boldsymbol{\nu}\tilde{m})\langle P \rangle Q \mid (\boldsymbol{\nu}\tilde{m})\langle E \rangle Q$

(π Output) $\dfrac{-}{\langle E \rangle \xrightarrow{\langle - \rangle} \langle E \rangle \mathbf{0}}$ (π Input) $\dfrac{-}{(x).P \xrightarrow{(E)} P\{^{E}/_{x}\}}$ (π Path) $\dfrac{E.(F.P) \xrightarrow{\pi} Q}{(E.F).P \xrightarrow{\pi} Q}$

(τ Eps) $\dfrac{-}{\epsilon.P \xrightarrow{\tau} P}$ (τ Comm) $\dfrac{P \xrightarrow{\langle - \rangle} (\boldsymbol{\nu}\tilde{m})\langle E \rangle P' \quad Q \xrightarrow{(E)} Q' \quad \mathrm{fn}(Q') \cap \{\tilde{m}\} = \emptyset}{P \mid Q \xrightarrow{\tau} (\boldsymbol{\nu}\tilde{m})(P' \mid Q')}$

Q. In Table 10 we give the rules that should be added to those of Table 4 and Table 5 to define the LTS for the closed processes of the extended calculus. Note that in the structural rules of Table 4 we are now assuming that parallel composition and restriction distribute over the new form of concretions $(\boldsymbol{\nu}\tilde{m})\langle E \rangle Q$ in the same manner as $(\boldsymbol{\nu}\tilde{m})\langle P \rangle Q$. The pre-action for output allows a uniform treatment of extrusion of names. Definition 3.2 and the extended LTS induce a bisimilarity relation, still denoted by $\approx$, over the closed systems of the message passing calculus.

We define the *open extension* $\mathcal{R}^{\mathrm{o}}$ of a relation $\mathcal{R}$ as: $P \; \mathcal{R}^{\mathrm{o}} \; Q$ if and and only if for every closing substitution σ mapping from variables to expressions, we have $P\sigma \; \mathcal{R} \; Q\sigma$.

Theorem 6.1 (Characterisation of $\cong^{\mathrm{o}}_{\mathrm{s}}$) *Relations $\approx^{\mathrm{o}}$ and $\cong^{\mathrm{o}}_{\mathrm{s}}$ coincide over systems in the message-passing calculus.*

Proof The extension of Theorem 3.8 (soundness of bisimilarity) to the message-passing calculus is straightforward. The extension of Theorem 3.16 (completeness of bisimilarity) follows because these relations are defined over systems and communication cannot be observed at top-level. □

The open extension of the relation $\mathcal{S}$, written $\mathcal{S}^{\mathrm{o}}$ can be shown equivalent to the relation

$$\mathcal{S}^{\mathrm{o}} = \{(P,Q) : k[P \mid R] \approx^{\mathrm{o}} k[Q \mid R], \text{ for all } k, R \text{ closed}\} \ .$$

Our characterisation of reduction barbed equivalence over processes lifts smoothly to the message passing calculus.

Theorem 6.2 (Characterisation of $\cong_{\mathrm{p}}^{\mathrm{e\,o}}$) *The relations $\cong_{\mathrm{p}}^{\mathrm{e\,o}}$ and $\mathcal{S}^{\mathrm{o}}$ coincide over processes in the message-passing calculus.*

Proof It is a easy extension of the proof of Theorem 5.3 to the closed terms of the message passing calculus. The result then follows from the definition of open extension. □

The context lemma can be rephrased for the message passing calculus.

Theorem 6.3 *Relations $\cong_{\mathrm{p}}^{\mathrm{e\,o}}$ and $\cong_{\mathrm{p}}^{\mathrm{o}}$ coincide over processes in the message-passing calculus.*

Proof The proof is an extension of the proof in the case without communication. We detail the case of closure under input prefix and replicated input prefix (for all the other cases it is enough to consider close terms).

Suppose that $P \cong_{\mathrm{p}}^{\mathrm{e\,o}} Q$ and that $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \subseteq \{x\}$. We want to show that $(x).P \cong_{\mathrm{p}}^{\mathrm{e}} (x).Q$. For that we use our characterisation of $\cong_{\mathrm{p}}^{\mathrm{e}}$ and we prove that for all n, R closed it holds $n[(x).P \mid R] \approx n[(x).Q \mid R]$. In particular, we prove that the relation

$$\mathcal{R} = \{(n[(x).P \mid R], n[(x).Q \mid R]) \ : \ P \cong_{\mathrm{p}}^{\mathrm{e\,o}} Q, \ \mathrm{fn}(P) \cup \mathrm{fn}(Q) \subseteq \{x\}, \text{ for all } n, R \text{ closed}\}^{=} \cup \ \approx$$

is a bisimulation up to context and up to structural congruence. The most interesting case is when $n[(x).P \mid R] \xrightarrow{\tau} n[(\boldsymbol{\nu}\tilde{r})(P\{^{E}/_{x}\} \mid R')] \equiv (\boldsymbol{\nu}\tilde{r})n[P\{^{E}/_{x}\} \mid R']$, where $n \notin \tilde{r}$. Observe that R sends the message E and resumes as R'. So we have a matching transition $n[(x).Q \mid R] \xrightarrow{\tau} \equiv (\boldsymbol{\nu}\tilde{r})n[Q\{^{E}/_{x}\} \mid R']$. Since $P \cong_{\mathrm{p}}^{\mathrm{e\,o}} Q$, it holds $P\{^{E}/_{x}\} \cong_{\mathrm{p}}^{\mathrm{e}} Q\{^{E}/_{x}\}$. The characterisation of $\cong_{\mathrm{p}}^{\mathrm{e}}$ guarantees that $n[P\{^{E}/_{x}\} \mid R'] \approx n[Q\{^{E}/_{x}\} \mid R']$ and this allows us to conclude that up to context we are still in $\mathcal{R}$.

Suppose that $P \cong_{\mathrm{p}}^{\mathrm{e\,o}} Q$ and that $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \subseteq \{x\}$. Now we want to show that $!(x).P \cong_{\mathrm{p}}^{\mathrm{e}} !(x).Q$. Reasoning as before, we prove that for all n, R closed it holds $n[!(x).P \mid R] \approx n[!(x).Q \mid R]$. In particular, we prove that the relation

$$\mathcal{R} = \{(n[!(x).P \mid R], n[!(x).Q \mid R)] \ : \ P \cong_{\mathrm{p}}^{\mathrm{e\,o}} Q, \ \mathrm{fn}(P) \cup \mathrm{fn}(Q) \subseteq \{x\}, \text{ for all } n, R \text{ closed}\}^{=} \cup \ \approx$$

is a bisimulation up to context and up to $\gtrsim\approx$. The most interesting case is when $n[!(x).P \mid R] \xrightarrow{\tau} n[(\boldsymbol{\nu}\tilde{r})(P\{^{E}/_{x}\} \mid !(x).P \mid R)] \equiv (\boldsymbol{\nu}\tilde{r})n[P\{^{E}/_{x}\} \mid !(x).P] \mid R'$, where $n \notin \tilde{r}$ and $\tilde{r} \cap \mathrm{fn}(P) = \emptyset$. Observe that R sends the message E and resumes as R'. So we have a matching transition $n[!(x).Q \mid R] \xrightarrow{\tau} \equiv (\boldsymbol{\nu}\tilde{r})n[Q\{^{E}/_{x}\} \mid !(x).Q \mid R']$, where $\tilde{r} \cap \mathrm{fn}(Q) = \emptyset$. By construction of $\mathcal{R}$ we have $n[P\{^{E}/_{x}\} \mid !(x).P \mid R'] \ \mathcal{R} \ n[P\{^{E}/_{x}\} \mid !(x).Q \mid R']$. Since $P \cong_{\mathrm{p}}^{\mathrm{e\,o}} Q$, it holds $P\{^{E}/_{x}\} \cong_{\mathrm{p}}^{\mathrm{e}} Q\{^{E}/_{x}\}$. The characterisation of $\cong_{\mathrm{p}}^{\mathrm{e}}$ guarantees that $n[P\{^{E}/_{x}\} \mid !(x).Q \mid R'] \approx n[Q\{^{E}/_{x}\} \mid !(x).Q \mid R']$. Since bisimilarity is closed under restriction we have $(\boldsymbol{\nu}\tilde{r})n[P\{^{E}/_{x}\} \mid !(x).Q \mid R'] \approx (\boldsymbol{\nu}\tilde{r})n[Q\{^{E}/_{x}\} \mid !(x).Q \mid R']$. This allows us to conclude that up to context (we factor out the context $(\boldsymbol{\nu}\tilde{r})(-)$) and up to $\gtrsim, \approx$ we are still in $\mathcal{R}$. □

Corollary 6.4 *Relations $\mathcal{S}^{\mathrm{o}}$ and $\cong_{\mathrm{p}}^{\mathrm{o}}$ coincide over processes in the message-passing calculus.*

A crucial aspect of working with systems deserves to be pointed out. Bisimilarity is defined over systems, and as such it cannot directly observe the exercise of communications capabilities (apart from internal communications). This allow us to avoid any special treatment for asynchronous communication. More than that, we can easily extend our results to a calculus equipped with *synchronous* communication (e.g., $\langle E\rangle.P$).

7 Algebraic Properties

In this section we prove a collection of algebraic laws using our bisimulation proof methods. Then, we prove the correctness of a protocol for controlling access through a *firewall*, first proposed in [6].

Laws on systems We briefly comment on the laws of Theorem 7.1. We recall that M, N range over systems and P, Q, R over processes. The first two laws are two examples of local communication within private ambients without interference. The third law is the well-known perfect firewall law. The following four laws represent non-interference properties about movements of private ambients. Finally, the last two laws say when opening cannot be interfered.

Theorem 7.1

1. $(\boldsymbol{\nu} n)n[\langle W\rangle.P \mid (x).Q \mid M] \;\cong_{\mathrm{p}}\; (\boldsymbol{\nu} n)n[P \mid Q\{^{W}/_{x}\} \mid M]$ *if* $n \notin \mathrm{fn}(M)$
2. $(\boldsymbol{\nu} n)n[\langle W\rangle.P \mid (x).Q \mid \prod_{j\in J} \mathtt{open}\_k_j.R_j] \;\cong_{\mathrm{p}}\; (\boldsymbol{\nu} n)n[P \mid Q\{^{W}/_{x}\} \mid \prod_{j\in J} \mathtt{open}\_k_j.R_j]$
3. $(\boldsymbol{\nu} n)n[P] \;\cong_{\mathrm{p}}\; \mathbf{0}$ *if* $n \notin \mathrm{fn}(P)$
4. $(\boldsymbol{\nu} n)((\boldsymbol{\nu} m)m[\mathtt{in}\_n.P] \mid n[M]) \;\cong_{\mathrm{p}}\; (\boldsymbol{\nu} n)n[(\boldsymbol{\nu} m)m[P] \mid M]$ *if* $n \notin \mathrm{fn}(M)$
5. $(\boldsymbol{\nu} m, n)(m[\mathtt{in}\_n.P] \mid n[\prod_{j\in J} \mathtt{open}\_k_j.R_j]) \;\cong_{\mathrm{p}}\; (\boldsymbol{\nu} m, n)n[m[P] \mid \prod_{j\in J} \mathtt{open}\_k_j.R_j]$
6. $(\boldsymbol{\nu} n)n[(\boldsymbol{\nu} m)m[\mathtt{out}\_n.P] \mid M] \;\cong_{\mathrm{p}}\; (\boldsymbol{\nu} n)((\boldsymbol{\nu} m)m[P] \mid n[M])$ *if* $n \notin \mathrm{fn}(M)$
7. $(\boldsymbol{\nu} n)n[m[\mathtt{out}\_n.P] \mid \prod_{j\in J} \mathtt{open}\_k_j.R_j] \;\cong_{\mathrm{p}}\; (\boldsymbol{\nu} n)(m[P] \mid n[\prod_{j\in J} \mathtt{open}\_k_j.R_j])$
   if $m \neq k_j$, *for* $j \in J$
8. $n[(\boldsymbol{\nu} m)(\mathtt{open}\_m.P \mid m[N]) \mid Q] \;\cong_{\mathrm{p}}\; n[(\boldsymbol{\nu} m)(P \mid N) \mid Q]$
   if $Q \equiv M \mid \prod_{j\in J}\langle W_j\rangle.R_j$ *and* $m \notin \mathrm{fn}(N)$
9. $(\boldsymbol{\nu} n)n[(\boldsymbol{\nu} m)(\mathtt{open}\_m.P \mid m[Q]) \mid R] \;\cong_{\mathrm{p}}\; (\boldsymbol{\nu} n)n[(\boldsymbol{\nu} m)(P \mid Q) \mid R]$
   if $R \equiv \prod_{i\in I}\langle W_i\rangle.S_i \mid \prod_{j\in J} \mathtt{open}\_k_j.R_j$ *and* $m, n \notin \mathrm{fn}(Q)$.

Proof To prove the above laws, except (3) and (9), we exhibit a bisimulation that relates them: the results will follow from Theorem 5.10. In all cases the bisimulation follows a similar pattern:

$$\mathcal{S} = \{(\mathit{lhs}, \mathit{rhs})\}^{=} \cup \approx$$

where *lhs* and *rhs* denote respectively the left hand side and the right hand side of the equation, parameterised over names, processes and systems. For proving the laws (3) and (9) we show that the above $\mathcal{S}$ is a bisimulation up to context and up to structural congruence. We illustrate

the proof of the law (3). Let $\mathcal{S} = \{((\boldsymbol{\nu} n)n[Q], \mathbf{0}) \mid \forall Q \text{ s.t. } n \notin \mathrm{fn}(Q)\}^{=}$. We show that $\mathcal{S}$ is a bisimulation up to context and up to structural congruence. The most delicate cases are those regarding the silent moves $*.\mathtt{enter}\_k$ and $*.\mathtt{exit}\_k$. For instance, if

$$(\boldsymbol{\nu} n)n[P] \xrightarrow{*.\mathtt{enter}\_k} (\boldsymbol{\nu} n)k[\circ \mid n[P']] \equiv k[\circ \mid (\boldsymbol{\nu} n)n[P']]$$

then

$$\mathbf{0} \mid k[\,\circ\,] \Rightarrow\equiv k[\circ \mid \mathbf{0}]$$

and up to context and structural congruence we are still in $\mathcal{S}$. □

Laws on processes In Theorem 7.2 we give a collection of algebraic laws involving processes. Law 1 says that the opening of private ambients, possibly containing arbitrary messages, cannot be observed. Law 2 says that stuttering is not observable as well. Law 3 shows that processes prefixed by private capabilities are garbage. Law 4 says that two processes that differ only for having received different private capabilities cannot be distinguished. An instance of this law is

$$(\boldsymbol{\nu} n)\langle C_n\rangle \cong_{\mathrm{p}} (\boldsymbol{\nu} n)\langle D_n\rangle$$

for $C_n, D_n \in \{\mathtt{in}\_n, \mathtt{out}\_n, \mathtt{open}\_n\}$. Notice that the above private outputs are not equivalent to $\mathbf{0}$ (use context $(x).a[\,]$, for a fresh). Law 5 is the Mobile Ambient variant of the *asynchrony law* [1] due to asynchronous communication. Finally, Law 6 equates two different outputs by adding a special process. While this law reminds us of Honda and Yoshida's *equator* [18], it should be pointed out that Honda and Yoshida's equators hide the difference between two channels, whereas we equate messages.

Theorem 7.2 (Process Laws)

1. $(\boldsymbol{\nu} n)(n[\prod_{j\in J}\langle E_j\rangle] \mid \mathtt{open}\_n.P) \;\cong_{\mathrm{p}}\; \prod_{j\in J}\langle E_j\rangle \mid P$ *if* $n \notin \mathrm{fn}(P, E_j)$ *for all* j
2. $\mathtt{in}\_n.\mathtt{out}\_n.\mathtt{in}\_n.P \;\cong_{\mathrm{p}}\; \mathtt{in}\_n.\mathtt{out}\_n.\mathtt{in}\_n.P \oplus \mathtt{in}\_n.P$ *where* $\oplus$ *is internal choice*
3. $(\boldsymbol{\nu} n)C_n.P \;\cong_{\mathrm{p}}\; \mathbf{0}$ *if* $C_n \in \{\mathtt{in}\_n, \mathtt{out}\_n, \mathtt{open}\_n\}$*;*
4. $(\boldsymbol{\nu} n)P\{^{C_n}\!/_x\} \;\cong_{\mathrm{p}}\; (\boldsymbol{\nu} n)P\{^{D_n}\!/_x\}$ *if* $C_n, D_n \in \{\mathtt{in}\_n, \mathtt{out}\_n, \mathtt{open}\_n\}$, $\mathrm{fv}(P) \subseteq \{x\}$, *and* $n \notin \mathrm{fn}(P)$.
5. $(x).\langle x\rangle \;\cong_{\mathrm{p}}\; \mathbf{0}$
6. $\langle E\rangle \mid \mathsf{Eq}(E, F) \;\cong_{\mathrm{p}}\; \langle F\rangle \mid \mathsf{Eq}(E, F)$ *where* $\mathsf{Eq}(E, F) \stackrel{\mathrm{def}}{=} !(x).\langle E\rangle \mid !(x).\langle F\rangle$

Proof By Theorems 5.2 and 5.3, it suffices to show that

$$k[lhs \mid R] \approx k[rhs \mid R]$$

for all k and R, where lhs and rhs denote the left hand side, right hand side, of each law. In all cases, except 4, this can be proved by showing that the relation

$$\mathcal{R}= \{(k[lhs \mid R], k[rhs \mid R]) : \text{ for all } k \text{ and } R\}^{=} \cup \mathcal{I}$$

is a bisimulation up to context and up to $\gtrsim \approx$, where $\mathcal{I}$ represent the identity relation over systems.

In Law 4, the equality to prove is $k[(\boldsymbol{\nu} n)P\{^{C_n\!/\!x}\} \mid R] \approx k[(\boldsymbol{\nu} n)P\{^{D_n\!/\!x}\} \mid R]$, for all k and R. This can be proved by showing that the relation

$$\mathcal{R}= \{((\boldsymbol{\nu} n)M\{^{C_n\!/\!x}\}, (\boldsymbol{\nu} n)M\{^{D_n\!/\!x}\}) : \text{fv}(M) \subseteq \{x\} \text{ and } n \notin \text{fn}(M)\}^{=}$$

is a bisimulation. Notice that, as R is closed, up to α-conversion, to avoid name-capturing, we have $k[(\boldsymbol{\nu} n)P\sigma \mid R] \equiv (\boldsymbol{\nu} n)k[P \mid R]\sigma$. □

On stuttering In [34] it is argued that barbed equivalences are insensitive to *stuttering* phenomena, originated by processes that may repeatedly enter and exit an ambient. Using a sum operator *à la* CCS, the next example conveys some intuitions about stuttering. The systems

$$M = m[\texttt{in\_}n.\texttt{out\_}n.\texttt{in\_}n.R] \qquad \text{and} \qquad N = m[\texttt{in\_}n.\texttt{out\_}n.\texttt{in\_}n.R + \texttt{in\_}n.R]$$

are indeed reduction barbed congruent. To see why the extra summand of N does not affect its behaviour, consider a reduction produced by this summand:

$$N \mid n[S] \;\rightarrow\!\!\!\!\circ\; n[S \mid m[R]] \;.$$

The process M can match it using three reductions:

$$M \mid n[S] \;\rightarrow\!\!\!\!\circ\; n[S \mid m[\texttt{out\_}n.\texttt{in\_}n.R]] \;\rightarrow\!\!\!\!\circ\; n[S] \mid m[\texttt{in\_}n.R] \;\rightarrow\!\!\!\!\circ\; n[S \mid m[R]] \;.$$

The crucial point is that the exercise of the capability $\texttt{in\_}n$ is matched by the exercise of three capabilities, $\texttt{in\_}n.\texttt{out\_}n.\texttt{in\_}n$. Although it might seem that our bisimilarity matches each action with only one action (possibly preceded and/or followed by τ transitions), our bisimilarity is actually insensitive to stuttering. To illustrate why, we use a variant of the example above that does not rely on internal sum. Replication in the processes P and Q below implements a loop with an alternation between input/output and the path $\texttt{in\_}n.\texttt{out\_}n$. There is a 1-cycle shift, however, between the two loops. Stuttering makes the shift irrelevant.

Proposition 7.3 *The processes P and Q defined as*

$$P = (\boldsymbol{\nu} l)(\texttt{in\_}n.l[] \mid !\texttt{open\_}l.\texttt{out\_}n.\texttt{in\_}n.l[])$$
$$Q = (\boldsymbol{\nu} l)(\texttt{in\_}n.\texttt{out\_}n.\texttt{in\_}n.l[] \mid !\texttt{open\_}l.\texttt{out\_}n.\texttt{in\_}n.l[])$$

are reduction barbed congruent over processes.

Proof Let

$$\begin{aligned}\mathcal{R} = \{(\; & k[O \mid (\boldsymbol{\nu} l)(\texttt{in\_}n.l[] \mid !\texttt{open\_}l.\texttt{out\_}n.\texttt{in\_}n.l[])] \;, \\ & k[O \mid (\boldsymbol{\nu} l)(\texttt{in\_}n.\texttt{out\_}n.\texttt{in\_}n.l[] \mid !\texttt{open\_}l.\texttt{out\_}n.\texttt{in\_}n.l[])] \;) \\ & \qquad \mid \; k \text{ and } O \text{ are arbitrary}\}^{=} \;\cup\; \mathcal{I} \;.\end{aligned}$$

where $\mathcal{I}$ is the identity relation between systems. The relation $\mathcal{R}$ is a bisimulation up to context and up to structural congruence. We detail the most interesting case, where the exercise of one capability must be matched by the exercise of three capabilities. Suppose $M \;\mathcal{R}\; N$, with

$$M = k[O \mid (\boldsymbol{\nu} l)(\texttt{in\_}n.l[] \mid !\texttt{open\_}l.\texttt{out\_}n.\texttt{in\_}n.l[])]$$

and

$$N = k[O \mid (\boldsymbol{\nu} l)(\texttt{in\_}n.\texttt{out\_}n.\texttt{in\_}n.l[] \mid !\texttt{open\_}l.\texttt{out\_}n.\texttt{in\_}n.l[])] \ .$$

Also suppose that

$$M \xrightarrow{k.\texttt{enter\_}n} n[\circ \mid k[O \mid (\boldsymbol{\nu} l)(l[] \mid !\texttt{open\_}l.\texttt{out\_}n.\texttt{in\_}n.l[])]] \ .$$

Then N can perform the following sequence of transitions:

$$\begin{aligned} N &\xrightarrow{k.\texttt{enter\_}n} n[\circ \mid k[O \mid (\boldsymbol{\nu} l)(\texttt{out\_}n.\texttt{in\_}n.l[] \mid !\texttt{open\_}l.\texttt{out\_}n.\texttt{in\_}n.l[])]] \\ &\xrightarrow{\tau} n[\,\circ\,] \mid k[O \mid (\boldsymbol{\nu} l)(\texttt{in\_}n.l[] \mid !\texttt{open\_}l.\texttt{out\_}n.\texttt{in\_}n.l[])] \\ &\xrightarrow{\tau} n[\circ \mid k[O \mid (\boldsymbol{\nu} l)(l[] \mid !\texttt{open\_}l.\texttt{out\_}n.\texttt{in\_}n.l[])]] \ . \end{aligned}$$

For all instantiations of $\circ$ we can factor out the context $n[\circ \mid -]$ and up to context we are still in $\mathcal{R}$. □

The proof above clearly shows how the exercise of the three capabilities $\texttt{in\_}n.\texttt{out\_}n.\texttt{in\_}n$ needed to match the capability $\texttt{in\_}n$ give rise to a $k.\texttt{enter\_}n$ action followed by two internal transitions. The internal actions are subsequently absorbed by the weak formulation of the equivalence.

Crossing a firewall A protocol is discussed in [6] for controlling access through a firewall. The ambient w represents the firewall; the ambient m, a trusted agent containing a process Q that is supposed to cross the firewall. The firewall ambient sends into the agent a pilot ambient k with the capability $\texttt{in\_}w$ for entering the firewall. The agent acquires the capability by opening k. The process Q carried by the agent is finally liberated inside the firewall by the opening of ambient m. Names m and k act like passwords which guarantee the access only to authorised agents. Here is the protocol in MA:

$$\begin{aligned} AG &\stackrel{\text{def}}{=} m[\texttt{open\_}k.(x).x.Q] \\ FW &\stackrel{\text{def}}{=} (\boldsymbol{\nu} w)w[\texttt{open\_}m.P \mid k[\texttt{out\_}w.\texttt{in\_}m.\langle \texttt{in\_}w \rangle]] \end{aligned}$$

The correctness (of a mild variant) of the protocol above is shown in [6] for may-testing [8] proving that

$$(\boldsymbol{\nu} m, k)(AG \mid FG) \ \cong_{\text{p}} \ (\boldsymbol{\nu} w)w[Q \mid P]$$

under the conditions that $w \notin \text{fn}(Q)$, $x \notin \text{fv}(Q)$, $\{m, k\} \cap (\text{fn}(P) \cup \text{fn}(Q)) = \emptyset$. The proof relies on non-trivial preserved by system contexts reasonings. In what follows, we show how it can be established using our bisimulation proof methods.

The system on the right can be obtained from that one on the left by executing six τ-actions. So, it suffices to prove that $\cong_{\text{p}}$ is insensitive to all these τ-actions. The result follows from the algebraic laws of Theorem 7.1 and the following two laws:

Lemma 7.4 *Let P, Q, and R be processes. Then*

1. $(\boldsymbol{\nu} k, m, w)(k[\texttt{in\_}m.P] \mid m[\texttt{open\_}k.Q] \mid w[\texttt{open\_}m.R])$
   $\cong_{\text{p}} \ (\boldsymbol{\nu} k, m, w)(m[k[P] \mid \texttt{open\_}k.Q] \mid w[\texttt{open\_}m.R])$
2. $(\boldsymbol{\nu} m, w)(m[\langle \texttt{in\_}w \rangle \mid (x).P] \mid w[\texttt{open\_}m.Q])$
   $\cong_{\text{p}} \ (\boldsymbol{\nu} m, w)(m[P\{^{\texttt{in\_}w}/_{x}\}] \mid w[\texttt{open\_}m.Q])$

Proof By exhibiting the appropriate bisimulation. Again, in all cases the bisimulation has a similar form:

$$\mathcal{S} = \{(lhs, rhs)\}^{=} \cup \approx$$

where *lhs* and *rhs* denote respectively the left hand side and the right hand side of the equation. □

Theorem 7.5 *If* $w \notin \mathrm{fn}(Q)$, $x \notin \mathrm{fv}(Q)$, *and* $\{m, k\} \cap (\mathrm{fn}(P) \cup \mathrm{fn}(Q)) = \emptyset$, *then*

$$(\boldsymbol{\nu} m, k)(AG \mid FG) \cong_{\mathrm{p}} (\boldsymbol{\nu} w)w[Q \mid P].$$

Proof It suffices to apply the algebraic laws of Theorem 7.1 and Lemma 7.4. More precisely, we apply, in sequence, Law (7) of Theorem 7.1, Law (1) of Lemma 7.4, Law (9) of Theorem 7.1, Law (2) of Lemma 7.4, and Laws (5) and (9) of Theorem 7.1. □

8 Related Work

In this paper we study the behavioural theory of Cardelli and Gordon's Mobile Ambients.

A theory of Morris-style preserved by system contexts equivalence for Mobile Ambients has been developed by Gordon and Cardelli in [12]. However, although the theory is equipped with a context lemma which allows to consider only contexts of a particular form, we believe that the verification of algebraic laws still remain quite complicated. It should be noticed that all the laws proved in [12] relate processes that engage only in limited interactions with their context.

Higher-order LTSs for Mobile Ambients can be found in [5, 12, 42, 9]. But we are not aware of any form of bisimilarity defined using these LTSs. In [39], Sewell addresses the problem of uniformly deriving LTSs and bisimulation congruences from the reduction rules of a calculus. The transitions generated for a fragment of Mobile Ambients require the same universal quantifications on the content of the interacting ambient as ours. Sewell's techniques only apply to strong equivalences. A simple first-order LTS for MA without restriction is proposed by Sangiorgi in [34]. Using this LTS the author defines an *intensional* bisimilarity for MA that separates terms on the basis of their internal structure.

Recently, Jensen and Milner [19], based on previous work by Leifer and Milner [20], derived an LTS for Mobile Ambients, starting from an encoding of Mobile Ambients into Bigraphs. We conjecture that the resulting bisimilarity, when limited to visible actions, coincides with the bisimilarity presented in this paper. On the other hand their bisimilarity does not validate equations based on unobservable migrations, like the perfect firewall equations.

Our work is the natural prosecution of [23, 24] where an LTS and a labelled characterisation of reduction barbed congruence are given for a more handful variant of Levi and Sangiorgi's Safe Ambients, called SAP. The main differences with respect to [23] are the following:

- Unlike MA, the calculus SAP is equipped with co-capabilities and passwords; both features are essential to prove the characterisation result in SAP. On the other hand in MA, (i) the presence of grave interferences, (ii) the asynchrony nature of ambient migration, and (iii) the non-observability of secret ambients, make the behavioural theory much more involved.

- Our env-actions, unlike those in [23], are truly late, as they do not mention the process provided by the environment. We add such process *later*, when playing the bisimulation game. This approach has then been adopted in [24].

- Our actions for ambient's movement, unlike those in SAP, report the name of the migrating ambient. For instance, in $k.\texttt{enter}\_n$ we say that ambient k enters n. The knowledge of k is necessary to make the action observable for the environment. This is not needed in SAP, because movements can be observed by means of co-capabilities.

- Co-capabilities in SAP also allow the observation of the movement of an ambient whose name is private. As a consequence, the perfect firewall equation does not hold neither in SAP, nor in Safe Ambients. By contrast, in MA the movements of an ambient whose name is private cannot be observed. This is why the perfect firewall equation holds.

- Here, we enhance our proof methods with up-to expansion and up-to context proof techniques.

Note that, although the labelled bisimilarity is contextual, it is an effective proof technique, especially when coupled with the up-to expansion and up-to context proof-techniques. As an example, the proofs of the algebraic laws of Section 7 are very simple. This should be contrasted with the proofs based on contextual reasoning developed in [22, 12].

Finally, apart from [23], other forms of bisimilarity for higher-order distributed calculi, such as Distributed π-calculus [15], Seal [43], a Calculus for Mobile Resources [11], NBA [4], SafeDpi [14], Homer [40], and the Kell calculus [38] can be found in [13, 7, 11, 4, 14, 40, 38], but only [13, 11, 4, 14, 38] prove labelled characterisations of a contextually-defined program equivalence (in [40] completeness holds only for the strong equivalence).

Unyapoth and Sewell [41] take a different, more intensional approach to define an equivalence for Nomadic Pict. To establish correctness of a particular protocol, they identify a novel equivalence based on coupled simulation but tailored to accommodate code migration. Although this equivalence has many interesting properties, in particular it is a congruence, is not shown to coincide with any independent contextually defined equivalence.

Acknowledgements The authors would like to thank Vladimiro Sassone who spotted a problem in the proof of Theorem 4.3 in an early draft of the paper. The second author is grateful to the Foundations of Computing Group of University of Sussex, for the kind hospitality and support.

References

[1] R. Amadio, I. Castellani, and D. Sangiorgi. On bisimulations for the asynchronous π-calculus. *Theoretical Computer Science*, 195:291–324, 1998.

[2] S. Arun-Kumar and M. Hennessy. An efficiency preorder for processes. *Acta Informatica*, 29:737–760, 1992.

[3] G. Boudol. Asynchrony and the π-calculus. Technical Report RR-1702, INRIA-Sophia Antipolis, 1992.

[4] M. Bugliesi, S. Crafa, M. Merro, and V. Sassone. Communication interference in mobile boxed ambients. Forthcoming Technical Report. An extended abstract appeared in Proc. FSTTCS'02, LNCS, Springer Verlag.

[5] L. Cardelli and A. Gordon. A commitment relation for the ambient calculus. Unpublished notes, 1996.

[6] L. Cardelli and A. Gordon. Mobile ambients. *Theoretical Computer Science*, 240(1):177–213, 2000. An extended abstract appeared in *Proc. of FoSSaCS '98*.

[7] G. Castagna and F. Zappa Nardelli. The seal calculus revisited: Contextual equivalence and bisimilarity. In *Proc. 22nd FSTTCS '02*, volume 2556 of *LNCS*. Springer Verlag, 2002.

[8] R. De Nicola and M. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1984.

[9] G. Ferrari, U. Montanari, and E. Tuosto. A LTS semantics of ambients via graph synchronization with mobility. In *Proc. ICTCS*, volume 2202 of *LNCS*, 2001.

[10] C. Fournet and G. Gonthier. A hierarchy of equivalences for asynchronous calculi. In *Proc. 25th ICALP*, pages 844–855. Springer Verlag, 1998.

[11] J.C. Godskesen, T. Hildebrandt, and V. Sassone. A calculus of mobile resources. In *Proc. 10th CONCUR '02*, volume 2421 of *LNCS*, 2002.

[12] A. D. Gordon and L. Cardelli. Equational properties of mobile ambients. *Journal of Mathematical Structures in Computer Science*, 12:1–38, 2002. An extended abstract appeared in *Proc. FoSSaCs '99*.

[13] M. Hennessy, M. Merro, and J. Rathke. Towards a behavioural theory of access and mobility control in distributed system. In *Proc. 5th FoSSaCS '03*, LNCS. Springer Verlag, 2003.

[14] M. Hennessy, J. Rathke, and N. Yoshida. Safedpi: A language for controlling mobile code. Computer Science Report 2003:02, University of Sussex, 2003. An extended abstract appeared in the Proc. FOSSACS'04, volume 2987, Lecture Notes in Computer Science. Springer-Verlag 2004.

[15] M. Hennessy and J. Riely. A typed language for distributed mobile processes. In *Proc. 25th POPL*. ACM Press, 1998.

[16] K. Honda and M. Tokoro. An Object Calculus for Asynchronous Communications. In *Proc. ECOOP '91*, volume 512 of *LNCS*. Springer Verlag, 1991.

[17] K. Honda and N. Yoshida. Replication in Concurrent Combinators. In *Proc. TACS'94*, volume 789 of *LNCS*. Springer Verlag, 1994.

[18] K. Honda and N. Yoshida. On reduction-based process semantics. *Theoretical Computer Science*, 152(2):437–486, 1995.

[19] O. H. Jensen and R. Milner. Bigraphs and mobile processes (revised). Technical Report 580, LFCS, Dept. of Comp. Sci., Edinburgh Univ., February 2004. An extended abstract appeared in *Conference Record of 30th Symposium on Principles of Programming Languages*, ACM Press, 2003.

[20] J. J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. In *CONCUR 2000 - Concurrency Theory, 11th International Conference, University Park, PA, USA, August 22–25, 2000, Proceedings*, volume 1877 of *LNCS*, pages 243–258. Springer-Verlag, 2000.

[21] F. Levi and D. Sangiorgi. Controlling interference in ambients. In *Proc. 27th POPL*. ACM Press, 2000.

[22] F. Levi and D. Sangiorgi. Controlling interference in ambients. An extended abstract appeared in *Proc. 27th Symposium on Principles of Programming Languages*, ACM Press, 2000.

[23] M. Merro and M. Hennessy. Bisimulation congruences in safe ambients. In *Proc. 29th POPL '02*. ACM Press, 2002.

[24] M. Merro and M. Hennessy. A bisimulation-based semantic theory for safe ambients. Submitted for publication, 2004.

[25] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[26] R. Milner. The polyadic π-calculus: a tutorial. Technical Report ECS–LFCS–91–180, LFCS, Dept. of Comp. Sci., Edinburgh Univ., October 1991. Also in *Logic and Algebra of Specification*, ed. F.L. Bauer, W. Brauer and H. Schwichtenberg, Springer Verlag, 1993.

[27] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, (Parts I and II). *Information and Computation*, 100:1–77, 1992.

[28] R. Milner and D. Sangiorgi. Barbed bisimulation. In *Proc. 19th ICALP*, volume 623 of *LNCS*, pages 685–695. Springer Verlag, 1992.

[29] D.M. Park. Concurrency on automata and infinite sequences. In P. Deussen, editor, *Conf. on Theoretical Computer Science*, volume 104 of *LNCS*. Springer Verlag, 1981.

[30] D. Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms*. PhD thesis CST–99–93, Dept. of Comp. Sci., Edinburgh Univ., 1992.

[31] D. Sangiorgi. Bisimulation for Higher-Order Process Calculi. *Information and Computation*, 131(2):141–178, 1996.

[32] D. Sangiorgi. Locality and non-interleaving semantics in calculi for mobile processes. *Theoretical Computer Science*, 155:39–83, 1996.

[33] D. Sangiorgi. On the bisimulation proof method. *Journal of Mathematical Structures in Computer Science*, 8:447–479, 1998.

[34] D. Sangiorgi. Extensionality and intensionality of the ambient logic. In *Proc. 28th POPL*. ACM Press, 2001.

[35] D. Sangiorgi and R. Milner. The problem of "Weak Bisimulation up to". In *Proc. CONCUR '92*, volume 630 of *LNCS*, pages 32–46. Springer Verlag, 1992.

[36] D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.

[37] D. Sangiorgi and D. Walker. Some results on barbed equivalences in pi-calculus. In *Proc. CONCUR '01*, volume 2154 of *LNCS*. Springer Verlag, 2001.

[38] A. Schmitt and J. Stefani. The kell calculus: A family of higher-order distributed process calculi. In *LNCS*. Springer-Verlag, To appear. Workshop of Global Computing 2004.

[39] P. Sewell. From rewrite rules to bisimulation congruences. *TCS*, 2003. To appear.

[40] M. Bundgaard T. Hildebrandt, J.C. Godskesen. Bisimulation congruences for homer. Technical Report TR-2004-52, ITU, 2004.

[41] A. Unyapoth and P. Sewell. Nomadic Pict: Correct communication infrastructures for mobile computation. In *Proc. 28th POPL*. ACM Press, 2001.

[42] M. G. Vigliotti. Transition systems for the ambient calculus. Master thesis, Imperial College of Science, Technology and Medicine (University of London), September 1999.

[43] J. Vitek and G. Castagna. Seal: A framework for secure mobile computations. In *Internet Programming Languages*, number 1686 in LNCS, pages 47–77. Springer Verlag, 1999.

A Proofs from Section 2

Proof of Theorem 2.3

Part 1. By induction on the derivation of $P \xrightarrow{\tau} P'$. Remark that τ-transitions can only be generated by the rules in Table 5.

(τ Enter) We know that $P \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{p})\langle P_1\rangle P_2$, and $Q \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{q})\langle Q_1\rangle Q_2$. From Lemma 2.2 we deduce that $P \equiv (\boldsymbol{\nu}\tilde{p})((\boldsymbol{\nu}\tilde{r})k[\texttt{in\_}n.P_3 \mid P_4] \mid P_2)$, where $P_1 \equiv (\boldsymbol{\nu}\tilde{r})k[P_3 \mid P_4]$, for some processes P_3, P_4 and names $\tilde{r}$. Lemma 2.2 also guarantees that $Q \equiv (\boldsymbol{\nu}\tilde{q})(n[Q_1] \mid Q_2)$. Then,

$$\begin{aligned} P \mid Q &\equiv (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{r})k[\texttt{in\_}n.P_3 \mid P_4] \mid P_2) \mid (\boldsymbol{\nu}\tilde{q})(n[Q_1] \mid Q_2) \\ &\equiv (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{r})(\boldsymbol{\nu}\tilde{q})(k[\texttt{in\_}n.P_3 \mid P_4] \mid n[Q_1] \mid P_2 \mid Q_2) \\ &\rightarrowtriangle (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{r})(\boldsymbol{\nu}\tilde{q})(n[k[P_3 \mid P_4] \mid Q_1] \mid P_2 \mid Q_2) \\ &\equiv (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{q})(n[P_1 \mid Q_1] \mid P_2 \mid Q_2) \end{aligned}$$

as desired.

(τ Exit) We know that $P \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{p})\langle k[P_1]\rangle P_2$. From Lemma 2.2 we deduce that $P \equiv (\boldsymbol{\nu}\tilde{p})((\boldsymbol{\nu}\tilde{r})k[\texttt{out\_}n.P_3 \mid P_4] \mid P_2)$, where $P_1 \equiv P_3 \mid P_4$, for some processes P_3, P_4 and names $\tilde{r}$. Then,

$$\begin{aligned} n[P] &\equiv n[(\boldsymbol{\nu}\tilde{p})((\boldsymbol{\nu}\tilde{r})k[\texttt{out\_}n.P_3 \mid P_4] \mid P_2)] \\ &\equiv (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{r})n[k[\texttt{out\_}n.P_3 \mid P_4] \mid P_2)] \\ &\rightarrowtriangle (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{r})(n[P_2]k[P_3 \mid P_4]) \\ &\equiv (\boldsymbol{\nu}\tilde{p})(n[P_2] \mid (\boldsymbol{\nu}\tilde{r})k[P_3 \mid P_4]) \end{aligned}$$

as desired.

(τ Open) We know that $P \xrightarrow{\texttt{open\_}n} P_1$ and $Q \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{q})\langle Q_1\rangle Q_2$. From Lemma 2.2 we deduce that $P \equiv (\boldsymbol{\nu}\tilde{p})(\texttt{open\_}n.P_2 \mid P_3)$, where $P_1 \equiv (\boldsymbol{\nu}\tilde{p})(P_2 \mid P_3)$ for for some processes P_2, P_3. Lemma 2.2 also guarantees that $Q \equiv (\boldsymbol{\nu}\tilde{q})(n[Q_1] \mid Q_2)$. Then,

$$\begin{array}{rcl} P \mid Q & \equiv & (\boldsymbol{\nu}\tilde{p})(\texttt{open\_}n.P_2 \mid P_3) \mid (\boldsymbol{\nu}\tilde{q})(n[Q_1] \mid Q_2) \\ & \equiv & (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{q})(\texttt{open\_}n.P_2 \mid n[Q_1] \mid P_3 \mid Q_2) \\ & \rightarrowtriangle & (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{q})(P_2 \mid Q_1 \mid P_3 \mid Q_2) \\ & \equiv & (\boldsymbol{\nu}\tilde{p})(P_2 \mid P_3) \mid (\boldsymbol{\nu}\tilde{q})(Q_1 \mid Q_2) \end{array}$$

as desired.

The other cases follows straightforwardly from the congruence rules of the reduction relation.

Part 2. By induction on the derivation of $P \rightarrowtriangle Q$. There are three base cases.

(Red In) We know that

$$n[\texttt{in\_}m.P \mid Q] \mid m[R] \;\rightarrowtriangle\; m[n[P \mid Q] \mid R]\ .$$

The derivation below is valid.

$$\dfrac{\dfrac{\dfrac{\texttt{in\_}m.P \xrightarrow{\texttt{in\_}m} P}{\texttt{in\_}m.P \mid Q \xrightarrow{\texttt{in\_}m} P \mid Q}}{n[\texttt{in\_}m.P \mid Q] \xrightarrow{\texttt{enter\_}m} \langle P \mid Q\rangle\mathbf{0}} \qquad m[R] \xrightarrow{\texttt{amb\_}m} \langle R\rangle\mathbf{0}}{n[\texttt{in\_}m.P \mid Q] \mid m[R] \;\xrightarrow{\tau}\; m[n[P \mid Q] \mid R]}$$

(Red Out) We know that

$$m[n[\texttt{out\_}m.P \mid Q] \mid R] \;\rightarrowtriangle\; n[P \mid Q] \mid m[R]$$

The derivation below is valid.

$$\dfrac{\dfrac{\dfrac{\dfrac{\texttt{out\_}m.P \xrightarrow{\texttt{out\_}m} P}{\texttt{out\_}m.P \mid Q \xrightarrow{\texttt{out\_}m} P \mid Q}}{n[\texttt{out\_}m.P \mid Q] \xrightarrow{\texttt{exit\_}m} \langle n[P \mid Q]\rangle\mathbf{0}}}{n[\texttt{out\_}m.P \mid Q] \mid R \xrightarrow{\texttt{exit\_}m} \langle n[P \mid Q]\rangle R}}{m[n[\texttt{out\_}m.P \mid Q] \mid R] \;\xrightarrow{\tau}\; n[P \mid Q] \mid m[R]}$$

(Red Open) We know that

$$\texttt{open\_}n.P \mid n[Q] \;\rightarrowtriangle\; P \mid Q$$

The derivation below is valid.

$$\dfrac{\texttt{open\_}n.P \xrightarrow{\texttt{open\_}n} P \qquad n[Q] \xrightarrow{\texttt{amb\_}n} \langle Q\rangle\mathbf{0}}{\texttt{open\_}n.P \mid n[Q] \;\xrightarrow{\tau}\; P \mid Q}$$

The induction step, rule (Red Struct), follows because τ-transitions are preserved by all static contexts. □

B Proofs from Section 3

Proof of Lemma 3.4

The relation $\mathcal{S}$ is preserved by system contexts, and as such it is the smallest relation between systems such that:

- if $M \mathcal{S} N$, then $M \mid H \mathcal{S} N \mid H$ for all systems H;
- if $M \mathcal{S} N$, then $(\boldsymbol{\nu} m)M \mathcal{S} (\boldsymbol{\nu} m)N$ for all names m;
- if $M \mathcal{S} N$, then $m[M \mid P] \mathcal{S} m[N \mid P]$ for all names m and processes P.

We prove the closure of $C[M] \mathcal{S} C[N]$ under the conditions for being a bisimulation by induction on the structure of $C[-]$.

- $C[-] = -$.

  This case holds because $M \mathcal{S} N$ satisfies the bisimulation conditions in SS.

- $C[-] = (\boldsymbol{\nu} m)D[-]$.

  We know that $D[M] \mathcal{S} D[N]$ satisfies the bisimulation conditions in $\mathcal{S}$, and we want to prove that $(\boldsymbol{\nu} m)D[M] \mathcal{S} (\boldsymbol{\nu} m)D[N]$ satisfies the bisimulation conditions in $\mathcal{S}$ as well. Suppose $(\boldsymbol{\nu} m)D[M] \xrightarrow{\alpha}$. We perform a case analysis on α.

  – $(\boldsymbol{\nu} m)D[M] \xrightarrow{\tau} O_1$.

  This can only be derived from $D[M] \xrightarrow{\tau} O_1$, where $O_1 = (\boldsymbol{\nu} m)O_1$. The induction hypothesis tells us that there exists a system O_2 such that $D[N] \Rightarrow O_2$ and $O_1 \mathcal{S} O_2$. We can derive $(\boldsymbol{\nu} m)D[N] \Rightarrow (\boldsymbol{\nu} m)O_2$ and conclude $(\boldsymbol{\nu} m)O_1 \mathcal{S} (\boldsymbol{\nu} m)O_2$ because $\mathcal{S}$ is closed under restriction.

  – $(\boldsymbol{\nu} m)D[M] \xrightarrow{k.\mathtt{enter}\_n} O_1$.

  Observe that this must have been derived from

$$\dfrac{\dfrac{D[M] \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{(\boldsymbol{\nu} m)D[M] \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}}{(\boldsymbol{\nu} m)D[M] \xrightarrow{k.\mathtt{enter}\_n} O_1 \equiv (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})(n[k[M_1] \mid \circ] \mid M_2)}$$

  for some process M_1 and system M_2. Remark that this implies $m \neq n$ and $m \neq k$. As $D[M] \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{k.\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})(n[k[M_1] \mid \circ] \mid M_2) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{k.\mathtt{enter}\_n} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P \mathcal{S} N' \bullet P$. As $A \xrightarrow{k.\mathtt{enter}\_n} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[k[N_1] \mid \circ] \mid N_2)$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$. This implies $(\boldsymbol{\nu} m)A \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$, from which we can derive $(\boldsymbol{\nu} m)A \xrightarrow{k.\mathtt{enter}\_n} C \equiv (\boldsymbol{\nu} m)B = (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})(n[k[N_1] \mid \circ] \mid N_2)$. We obtain $(\boldsymbol{\nu} m)D[N] \Rightarrow (\boldsymbol{\nu} m)A \xrightarrow{k.\mathtt{enter}\_n} C \Rightarrow\equiv (\boldsymbol{\nu} m)N'$. Call $(\boldsymbol{\nu} m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \mathcal{S} O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

– $(\boldsymbol{\nu} m)D[M] \xrightarrow{k.\mathtt{exit}\_n} O_1$.
Observe that this must have been derived from

$$\dfrac{\dfrac{D[M] \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{(\boldsymbol{\nu} m)D[M] \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}}{(\boldsymbol{\nu} m)D[M] \xrightarrow{k.\mathtt{exit}\_n} O_1 \;\equiv\; (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1])}$$

for some process M_1 and system M_2. Remark that this implies $m \neq n$ and $m \neq k$. As $D[M] \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{k.\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1]) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{k.\mathtt{exit}\_n} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$. As $A \xrightarrow{k.\mathtt{exit}\_n} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2] \mid k[N_1])$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$. This implies $(\boldsymbol{\nu} m)A \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$, from which we can derive $(\boldsymbol{\nu} m)A \xrightarrow{k.\mathtt{exit}\_n} C \equiv (\boldsymbol{\nu} m)B = (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2] \mid k[N_1])$. We obtain $(\boldsymbol{\nu} m)D[N] \Rightarrow (\boldsymbol{\nu} m)A \xrightarrow{k.\mathtt{exit}\_n} C \Rightarrow\equiv (\boldsymbol{\nu} m)N'$. Call $(\boldsymbol{\nu} m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \;\mathcal{S}\; O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

– $(\boldsymbol{\nu} m)D[M] \xrightarrow{n.\overline{\mathtt{enter}}\_k} O_1$.
Observe that this must have been derived from

$$\dfrac{\dfrac{D[M] \xrightarrow{\mathtt{amb}\_n} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2}{(\boldsymbol{\nu} m)D[M] \xrightarrow{\mathtt{amb}\_n} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2}}{(\boldsymbol{\nu} m)D[M] \xrightarrow{n.\overline{\mathtt{enter}}\_k} O_1 \;\equiv\; (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})(n[k[\,\circ\,] \mid M_1] \mid M_2)}$$

for some process M_1 and system M_2. Remark that this implies $m \neq n$ and $m \neq k$. As $D[M] \xrightarrow{\mathtt{amb}\_n} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2$ then $D[M] \xrightarrow{n.\overline{\mathtt{enter}}\_k} (\boldsymbol{\nu}\tilde{r})(n[k[\,\circ\,] \mid M_1] \mid M_2) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{n.\overline{\mathtt{enter}}\_k} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$. As $A \xrightarrow{n.\overline{\mathtt{enter}}\_k} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[k[\,\circ\,] \mid N_1] \mid N_2)$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\mathtt{amb}\_n} (\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2$. This implies $(\boldsymbol{\nu} m)A \xrightarrow{\mathtt{amb}\_n} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2$, from which we can derive $(\boldsymbol{\nu} m)A \xrightarrow{n.\overline{\mathtt{enter}}\_k} C \equiv (\boldsymbol{\nu} m)B = (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})(n[k[\,\circ\,] \mid N_1] \mid N_2)$. We obtain $(\boldsymbol{\nu} m)D[N] \Rightarrow (\boldsymbol{\nu} m)A \xrightarrow{n.\overline{\mathtt{enter}}\_k} C \Rightarrow\equiv (\boldsymbol{\nu} m)N'$. Call $(\boldsymbol{\nu} m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \;\mathcal{S}\; O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

– $(\boldsymbol{\nu} m)D[M] \xrightarrow{k.\mathtt{open}\_n} O_1$.

Observe that this must have been derived from

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2}{(\boldsymbol{\nu}m)D[M] \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2}}{(\boldsymbol{\nu}m)D[M] \xrightarrow{k.\texttt{open\_}n} O_1 \;\equiv\; k[\circ \mid (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2)]}$$

for some process M_1 and system M_2. Remark that this implies $m \neq n$ and $m \neq k$. As $D[M] \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2$ then $D[M] \xrightarrow{k.\texttt{open\_}n} k[\circ \mid (\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2)] = M'$. Also observe that $O_1 \equiv (\boldsymbol{\nu}m)k[\circ \mid (\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2)] = (\boldsymbol{\nu}m)M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{k.\texttt{open\_}n} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$. As $A \xrightarrow{k.\texttt{open\_}n} B$, the system B must be of the form $k[\circ \mid (\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2)]$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2$. This implies $(\boldsymbol{\nu}m)A \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2$, from which we can derive $(\boldsymbol{\nu}m)A \xrightarrow{k.\texttt{open\_}n} C \equiv k[\circ \mid (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2)] \equiv (\boldsymbol{\nu}m)k[\circ \mid (\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2)] = (\boldsymbol{\nu}m)N'$. We obtain $(\boldsymbol{\nu}m)D[N] \Rightarrow (\boldsymbol{\nu}m)A \xrightarrow{k.\texttt{open\_}n} C \Rightarrow\equiv (\boldsymbol{\nu}m)N'$. Call $(\boldsymbol{\nu}m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \;\mathcal{S}\; O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

– $(\boldsymbol{\nu}m)D[M] \xrightarrow{*.\texttt{enter\_}n} O_1$.
Observe that there are two possible derivations.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle m[M_1]\rangle M_2}{(\boldsymbol{\nu}m)D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{r})\langle m[M_1]\rangle M_2}}{(\boldsymbol{\nu}m)D[M] \xrightarrow{*.\texttt{enter\_}n} O_1 \;\equiv\; (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{r})(n[m[M_1] \mid \circ] \mid M_2)}$$

where $m \notin \tilde{r}$, for some process M_1 and system M_2. Remark that this implies $n \notin r$. As $D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle m[M_1]\rangle M_2$ then $D[M] \xrightarrow{m.\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})(n[m[M_1] \mid \circ] \mid M_2) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{m.\texttt{enter\_}n} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$. As $A \xrightarrow{m.\texttt{enter\_}n} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[m[N_1] \mid \circ] \mid N_2)$, for some process N_1 and system N_2, where $m \notin \tilde{s}$. It also holds $A \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{s})\langle m[N_1]\rangle N_2$. This implies $(\boldsymbol{\nu}m)A \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{s})\langle m[N_1]\rangle N_2$, from which we can derive $(\boldsymbol{\nu}m)A \mid n[\,\circ\,] \xrightarrow{\tau} C \equiv (\boldsymbol{\nu}m)B = (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{s})(n[m[N_1] \mid \circ] \mid N_2)$. We obtain $(\boldsymbol{\nu}m)(D[N] \mid n[\circ]) \equiv (\boldsymbol{\nu}m)D[N] \mid n[\circ] \Rightarrow (\boldsymbol{\nu}m)A \mid n[\circ] \xrightarrow{\tau} C \Rightarrow\equiv (\boldsymbol{\nu}m)N'$. Call $(\boldsymbol{\nu}m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \;\mathcal{S}\; O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{(\boldsymbol{\nu}m)D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}}{(\boldsymbol{\nu}m)D[M] \xrightarrow{*.\texttt{enter\_}n} O_1 \;\equiv\; (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{r})(n[k[M_1] \mid \circ] \mid M_2)}$$

where $k \in \tilde{r}$, for some process M_1 and system M_2. Remark that $n \notin \tilde{r}$. As $D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{*.\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})(n[k[M_1] \mid \circ] \mid M_2) = M'$. The induction hypothesis then tells us that there exist a system N' such that $D[N] \mid n[\,\circ\,] \Rightarrow N'$, and for all processes P it holds $M' \bullet P \mathcal{S} N' \bullet P$. We can derive $(\boldsymbol{\nu}m)D[N] \mid n[\,\circ\,] \equiv (\boldsymbol{\nu}m)(D[N] \mid n[\,\circ\,]) \Rightarrow (\boldsymbol{\nu}m)N'$. Call $(\boldsymbol{\nu}m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \mathcal{S} O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

– $(\boldsymbol{\nu}m)D[M] \xrightarrow{*.\texttt{exit\_}n} O_1$.

Observe that there are two possible derivations.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle m[M_1]\rangle M_2}{(\boldsymbol{\nu}m)D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{r})\langle m[M_1]\rangle M_2}}{(\boldsymbol{\nu}m)D[M] \xrightarrow{*.\texttt{exit\_}n} O_1 \equiv (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid m[M_1])}$$

where $m \notin \tilde{r}$, for some process M_1 and system M_2. Remark that this implies $n \notin r$. As $D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle m[M_1]\rangle M_2$ then $D[M] \xrightarrow{m.\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid m[M_1]) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{m.\texttt{exit\_}n} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P \mathcal{S} N' \bullet P$. As $A \xrightarrow{m.\texttt{exit\_}n} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2] \mid m[N_1])$, for some process N_1 and system N_2, where $m \notin \tilde{s}$. It also holds $A \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$. This implies $(\boldsymbol{\nu}m)A \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{s})\langle m[N_1]\rangle N_2$, from which we can derive $(\boldsymbol{\nu}m)n[\circ \mid A] \xrightarrow{\tau} C \equiv (\boldsymbol{\nu}m)B = (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2] \mid m[N_1])$. We obtain $(\boldsymbol{\nu}m)(D[N] \mid n[\,\circ\,]) \equiv (\boldsymbol{\nu}m)D[N] \mid n[\,\circ\,] \Rightarrow (\boldsymbol{\nu}m)A \mid n[\,\circ\,] \xrightarrow{\tau} C \Rightarrow\equiv (\boldsymbol{\nu}m)N'$. Call $(\boldsymbol{\nu}m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \mathcal{S} O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{(\boldsymbol{\nu}m)D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}}{(\boldsymbol{\nu}m)D[M] \xrightarrow{*.\texttt{exit\_}n} O_1 \equiv (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1])}$$

where $k \in \tilde{r}$, for some process M_1 and system M_2. Remark that $n \notin \tilde{r}$. As $D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{*.\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1]) = M'$. The induction hypothesis then tells us that there exist a system N' such that $n[\circ \mid D[N]] \Rightarrow N'$, and for all processes P it holds $M' \bullet P \mathcal{S} N' \bullet P$. We can derive $(\boldsymbol{\nu}m)D[N] \mid n[\circ] \equiv (\boldsymbol{\nu}m)(D[N] \mid n[\circ]) \Rightarrow (\boldsymbol{\nu}m)N'$. Call $(\boldsymbol{\nu}m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \mathcal{S} O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

• $C[-] = D[-] \mid H$.

We know that $D[M]\ \mathcal{S}\ D[N]$ satisfies the bisimulation conditions in $\mathcal{S}$, and we want to prove that $D[M] \mid H\ \mathcal{S}\ D[N] \mid H$ satisfies the bisimulation conditions in $\mathcal{S}$ as well. We perform a case analysis on the transition $D[M] \mid H \xrightarrow{\alpha} O_1$.

We consider first the cases when there is no interaction between $D[M]$ and H.

- $D[M] \mid H \xrightarrow{\tau} O_1$, because $D[M] \xrightarrow{\tau} M'$ and $O_1 \equiv M' \mid H$. The induction hypothesis tells us that there exists a N' such that $D[N] \Rightarrow N'$ and $M'\ \mathcal{S}\ N'$. Thus, $D[N] \mid H \Rightarrow O_2 \equiv N' \mid H$ and $O_1 \equiv M' \mid H\ \mathcal{S}\ N' \mid H \equiv O_2$ because $\mathcal{S}$ is closed under parallel composition.
- $D[M] \mid H \xrightarrow{\tau} O_1$, because $H \xrightarrow{\tau} H'$ and $O_1 \equiv D[M] \mid H'$. Let $O_2 = D[N] \mid H'$: it holds $D[N] \mid H \xrightarrow{\tau} O_2$, and $O_1\ \mathcal{S}\ O_2$ because $D[M]\ \mathcal{S}\ D[N]$ and $\mathcal{S}$ is closed under parallel composition.
- $D[M] \mid H \xrightarrow{k.\mathtt{enter}\_n} O_1$.
  There are two possible derivations.
  - Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{D[M] \mid H \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2 \mid H}}{D[M] \mid H \xrightarrow{k.\mathtt{enter}\_n} O_1 \ \equiv\ (\boldsymbol{\nu}\tilde{r})(n[k[M_1] \mid \circ] \mid M_2 \mid H)}$$

    for some process M_1 and system M_2. Remark that $k \notin \tilde{r}$. As $D[M] \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{k.\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})(n[k[M_1] \mid \circ] \mid M_2) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{k.\mathtt{enter}\_n} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P\ \mathcal{S}\ N' \bullet P$. As $A \xrightarrow{k.\mathtt{enter}\_n} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[k[N_1] \mid \circ] \mid N_2)$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$. This implies $A \mid H \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2 \mid H$, from which we can derive $A \mid H \xrightarrow{k.\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{s})(n[k[N_1] \mid \circ] \mid N_2 \mid H) \equiv B \mid H$. We obtain $D[N] \mid H \Rightarrow A \mid H \xrightarrow{k.\mathtt{enter}\_n} \equiv B \mid H \Rightarrow \equiv N' \mid H$. Call $N' \mid H = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P\ \mathcal{S}\ O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under parallel composition.
  - Suppose:

$$\dfrac{\dfrac{H \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[M] \mid H \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[M]}}{D[M] \mid H \xrightarrow{k.\mathtt{enter}\_n} O_1 \ \equiv\ (\boldsymbol{\nu}\tilde{r})(n[\circ \mid k[H_1]] \mid H_2 \mid M)}$$

    for some process H_1 and system H_2. Remark that $k \notin \tilde{r}$. We can construct the following derivation:

$$\dfrac{\dfrac{H \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[N] \mid H \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[N]}}{D[N] \mid H \xrightarrow{k.\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid k[H_1]] \mid H_2 \mid D[N]) \ =\ O_2}$$

We can conclude that for all processes P, it holds $O_1 \bullet P \; \mathcal{S} \; O_2 \bullet P$ up to structural congruence, because $D[M] \; \mathcal{S} \; D[N]$ and $\mathcal{S}$ is closed under parallel composition.

– $D[M] \mid H \xrightarrow{k.\mathtt{exit}\_n} O_1$.
There are two possible derivations.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{D[M] \mid H \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2 \mid H}}{D[M] \mid H \xrightarrow{k.\mathtt{exit}\_n} O_1 \;\equiv\; (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2 \mid H] \mid k[M_1])}$$

for some process M_1 and system M_2. Remark that $k \notin \tilde{r}$. As $D[M] \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{k.\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1]) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{k.\mathtt{exit}\_n} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P \; \mathcal{S} \; N' \bullet P$. Remark that $N' \equiv (\boldsymbol{\nu}\tilde{h})n[\circ \mid N_3] \mid N_4$, for some N_3, N_4. As $A \xrightarrow{k.\mathtt{exit}\_n} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2] \mid k[N_1])$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$. This implies $A \mid H \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2 \mid H$, from which we can derive $A \mid H \xrightarrow{k.\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2 \mid H] \mid k[N_1]) \equiv B \bullet (\circ \mid H)$. We obtain $D[N] \mid H \Rightarrow A \mid H \xrightarrow{k.\mathtt{exit}\_n} B \bullet (\circ \mid H) \Rightarrow\equiv N' \bullet (\circ \mid H)$. Call $N' \bullet (\circ \mid H) = O_2$. As for all processes P it holds $M' \bullet P \; \mathcal{S} \; N' \bullet P$, we can conclude that for all processes Q, it holds $O_1 \bullet Q \; \mathcal{S} \; O_2 \bullet Q$ up to structural congruence, because $O_1 \bullet Q \equiv M' \bullet (Q \mid H) \; \mathcal{S} \; N' \bullet (Q \mid H) \equiv O_2 \bullet Q$.

∗ Suppose:

$$\dfrac{\dfrac{H \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[M] \mid H \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[M]}}{D[M] \mid H \xrightarrow{k.\mathtt{exit}\_n} O_1 \;\equiv\; (\boldsymbol{\nu}\tilde{r})(n[\circ \mid H_2 \mid D[M]] \mid k[H_1])}$$

for some process H_1 and system H_2. Remark that $k \notin \tilde{r}$. We can construct the following derivation:

$$\dfrac{\dfrac{H \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[N] \mid H \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[N]}}{D[N] \mid H \xrightarrow{k.\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid H_2 \mid D[N]] \mid k[H_1]) \;=\; O_2}$$

We can conclude that for all processes P, it holds $O_1 \bullet P \; \mathcal{S} \; O_2 \bullet P$ up to structural congruence, because $D[M] \; \mathcal{S} \; D[N]$ and $\mathcal{S}$ is closed under parallel composition and ambient.

– $D[M] \mid H \xrightarrow{n.\overline{\mathtt{enter}}\_k} O_1$.
There are two possible derivations.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2}{D[M] \mid H \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2 \mid H}}{D[M] \mid H \xrightarrow{n.\overline{\texttt{enter}}\_k} O_1 \;\equiv\; (\boldsymbol{\nu}\tilde{r})(n[k[\,\circ\,] \mid M_1] \mid M_2 \mid H)}$$

for some process M_1 and system M_2. Remark that $k, n \notin \tilde{r}$. As $D[M] \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2$ then $D[M] \xrightarrow{n.\overline{\texttt{enter}}\_k} (\boldsymbol{\nu}\tilde{r})(n[k[\,\circ\,] \mid M_1] \mid M_2) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{n.\overline{\texttt{enter}}\_k} B \Rightarrow N'$, and for all processes P it holds $M'\bullet P \;\mathcal{S}\; N'\bullet P$. As $A \xrightarrow{n.\overline{\texttt{enter}}\_k} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[k[\,\circ\,] \mid N_1] \mid N_2)$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2$. This implies $A \mid H \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2 \mid H$, from which we can derive $A \mid H \xrightarrow{n.\overline{\texttt{enter}}\_k} (\boldsymbol{\nu}\tilde{s})(n[k[\,\circ\,] \mid N_1] \mid N_2 \mid H) \equiv B \mid H$. We obtain $D[N] \mid H \Rightarrow A \mid H \xrightarrow{n.\overline{\texttt{enter}}\_k} \equiv B \mid H \Rightarrow \equiv N' \mid H$. Call $N' \mid H = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \;\mathcal{S}\; O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under parallel composition.

∗ Suppose:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{r})\langle H_1\rangle H_2}{D[M] \mid H \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{r})\langle H_1\rangle H_2 \mid D[M]}}{D[M] \mid H \xrightarrow{n.\overline{\texttt{enter}}\_k} O_1 \;\equiv\; (\boldsymbol{\nu}\tilde{r})(n[k[\,\circ\,] \mid H_1] \mid H_2 \mid D[M])}$$

for some process H_1 and system H_2. Remark that $k \notin \tilde{r}$. We can construct the following derivation:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{r})\langle H_1\rangle H_2}{D[N] \mid H \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{r})\langle H_1\rangle H_2 \mid D[N]}}{D[N] \mid H \xrightarrow{n.\overline{\texttt{enter}}\_k} (\boldsymbol{\nu}\tilde{r})(n[k[\,\circ\,] \mid H_1] \mid H_2 \mid D[N]) \;=\; O_2}$$

We can conclude that for all processes P, it holds $O_1\bullet P \;\mathcal{S}\; O_2\bullet P$ up to structural congruence, because $D[M] \;\mathcal{S}\; D[N]$ and $\mathcal{S}$ is closed under parallel composition.

– $D[M] \mid H \xrightarrow{k.\texttt{open}\_n} O_1$.

There are two possible derivations.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2}{D[M] \mid H \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2 \mid H}}{D[M] \mid H \xrightarrow{k.\texttt{open}\_n} O_1 \;\equiv\; k[\circ \mid (\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2) \mid H]}$$

for some process M_1 and system M_2. Remark that $k, n \notin \tilde{r}$. As $D[M] \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2$ then $D[M] \xrightarrow{k.\texttt{open\_}n} k[\circ \mid (\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2)]$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{k.\texttt{open\_}n} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P \mathcal{S} N' \bullet P$. As $A \xrightarrow{k.\texttt{open\_}n} B$, the system B must be of the form $k[\circ \mid (\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2)]$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2$. This implies $A \mid H \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2 \mid H$, from which we can derive $A \mid H \xrightarrow{k.\texttt{open\_}n} k[\circ \mid (\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2) \mid H] \equiv B \bullet (\circ \mid H)$. We obtain $D[N] \mid H \Rightarrow A \mid H \xrightarrow{k.\texttt{open\_}n} \equiv B \bullet (\circ \mid H) \Rightarrow \equiv N' \bullet (\circ \mid H)$. Call $N' \bullet (\circ \mid H) = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \mathcal{S} O_2 \bullet P$ up to structural congruence, because for all processes P it holds $M' \bullet (P \mid H) \mathcal{S} N' \bullet (P \mid H)$.

∗ Suppose:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{h})\langle H_1\rangle H_2}{D[M] \mid H \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{h})\langle H_1\rangle H_2 \mid D[M]}}{D[M] \mid H \xrightarrow{k.\texttt{open\_}n} O_1 \equiv k[\circ \mid (\boldsymbol{\nu}\tilde{h})(H_1 \mid H_2) \mid D[M]]}$$

for some process H_1 and system H_2. Remark that $k \notin \tilde{h}$. We can construct the following derivation:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{h})\langle H_1\rangle H_2}{D[N] \mid H \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{h})\langle H_1\rangle H_2 \mid D[N]}}{D[N] \mid H \xrightarrow{k.\texttt{open\_}n} k[\circ \mid (\boldsymbol{\nu}\tilde{h})(H_1 \mid H_2) \mid D[N]] = O_2}$$

We can conclude that for all processes P, it holds $O_1 \bullet P \mathcal{S} O_2 \bullet P$ up to structural congruence, because $D[M] \mathcal{S} D[N]$ and $\mathcal{S}$ is closed under parallel composition and ambient.

– $D[M] \mid H \xrightarrow{*.\texttt{enter\_}n} O_1$.

There are two possible derivations.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{D[M] \mid H \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2 \mid H}}{D[M] \mid H \xrightarrow{*.\texttt{enter\_}n} O_1 \equiv (\boldsymbol{\nu}\tilde{r})(n[k[M_1] \mid \circ] \mid M_2 \mid H)}$$

where $k \in \tilde{r}$, for some process M_1 and system M_2. Remark that $n \notin \tilde{r}$. As $D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{*.\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})(n[k[M_1] \mid \circ] \mid M_2) = M'$. The induction hypothesis then tells us that there exist a system N' such that $D[N] \mid n[\circ] \Rightarrow N'$, and for all processes P it holds $M' \bullet P \mathcal{S} N' \bullet P$. We can derive $D[N] \mid n[\circ] \mid H \Rightarrow N' \mid H$. Call $N' \mid H = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \mathcal{S} O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under parallel composition.

∗ Suppose:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[M] \mid H \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[M]}}{D[M] \mid H \xrightarrow{*\texttt{.enter\_}n} O_1 \;\equiv\; (\boldsymbol{\nu}\tilde{r})(n[k[H_1] \mid \circ] \mid H_2 \mid D[M])}$$

where $k \in \tilde{r}$ for some process H_1 and system H_2. We can construct the following derivation:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[N] \mid H \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[N]} \qquad n[\,\circ\,] \xrightarrow{\texttt{amb\_}n} \langle\,\circ\,\rangle \mathbf{0}}{D[N] \mid H \mid n[\,\circ\,] \xrightarrow{\tau} (\boldsymbol{\nu}\tilde{r})(n[k[H_1] \mid \circ] \mid H_2 \mid D[N]) \;=\; O_2}$$

We can conclude that for all processes P, it holds $O_1 \bullet P \;\mathcal{S}\; O_2 \bullet P$ up to structural congruence, because $D[M] \;\mathcal{S}\; D[N]$ and $\mathcal{S}$ is closed under parallel composition.

– $D[M] \mid H \xrightarrow{*\texttt{.exit\_}n} O_1$.
There are two possible derivations.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{D[M] \mid H \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2 \mid H}}{D[M] \mid H \xrightarrow{*\texttt{.exit\_}n} O_1 \;\equiv\; (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2 \mid H] \mid k[M_1])}$$

for some process M_1 and system M_2. Remark that $k \in \tilde{r}$. As $D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{*\texttt{.exit\_}n} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1]) = M'$. The induction hypothesis then tells us that there exist systems N' such that $n[\circ \mid D[N]] \;\Rightarrow\; N'$, and for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$. Remark that $N' \equiv (\boldsymbol{\nu}\tilde{s})n[\circ \mid N_3] \mid N_4$, for some N_3, N_4. We can derive $n[\,\circ D[N] \mid H] \;\Rightarrow\; (\boldsymbol{\nu}\tilde{s})n[\circ \mid N_3 \mid H] \mid N_4$. Call $(\boldsymbol{\nu}\tilde{s})n[\circ \mid N_3 \mid H] \mid N_4 = O_2$. As for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$, we can conclude that for all processes Q, it holds $O_1 \bullet Q \;\mathcal{S}\; O_2 \bullet Q$ up to structural congruence, because $O_1 \bullet Q \equiv M' \bullet (Q \mid H) \;\mathcal{S}\; N' \bullet (Q \mid H) \equiv O_2 \bullet Q$.

∗ Suppose:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[M] \mid H \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[M]}}{D[M] \mid H \xrightarrow{*\texttt{.exit\_}n} O_1 \;\equiv\; (\boldsymbol{\nu}\tilde{r})(n[\circ \mid H_2 \mid D[M]] \mid k[H_1])}$$

for some process H_1 and system H_2. Remark that $k \in \tilde{r}$. We can construct the following derivation:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[N] \mid H \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[N]}}{n[\circ \mid D[N] \mid H] \xrightarrow{\tau} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid H_2 \mid D[N]] \mid k[H_1]) = O_2}$$

We can conclude that for all processes P, it holds $O_1 \bullet P \; \mathcal{S} \; O_2 \bullet P$ up to structural congruence, because $D[M] \; \mathcal{S} \; D[N]$ and $\mathcal{S}$ is closed under parallel composition and ambient.

Then, we consider the cases when there is interaction between $D[M]$ and H.

– $D[M] \mid H \xrightarrow{\tau} O_1$, because

$$D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{m})\langle k[M_1]\rangle M_2 \text{ and } H \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{h})\langle H_1\rangle H_2.$$

Then $O_1 \equiv (\boldsymbol{\nu}\tilde{h}, \tilde{m})(n[k[M_1] \mid H_1] \mid M_2 \mid H_2)$. We distinguish the cases $k \in \tilde{m}$, and $k \notin \tilde{m}$.

∗ $k \notin \tilde{m}$. As $D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{m})\langle k[M_1]\rangle M_2$, it also holds $D[M] \xrightarrow{k.\texttt{enter\_}n} M' \equiv (\boldsymbol{\nu}\tilde{m})(n[k[M_1] \mid \circ] \mid M_2)$. The induction hypothesis tells us that there exists a system N' such that $D[N] \stackrel{k.\texttt{enter\_}n}{\Longrightarrow} N' \equiv (\boldsymbol{\nu}\tilde{m})(n[k[N_1] \mid \circ] \mid N_2)$, and for all processes P, it holds $M' \bullet P \; \mathcal{S} \; N' \bullet P$. But if $D[N] \stackrel{k.\texttt{enter\_}n}{\Longrightarrow} N'$, then $D[N] \stackrel{\texttt{enter\_}n}{\Longrightarrow} (\boldsymbol{\nu}\tilde{m})\langle k[N_1]\rangle N_2$. This implies that $D[N] \mid H \stackrel{\tau}{\Longrightarrow} O_2 \equiv (\boldsymbol{\nu}\tilde{h}, \tilde{n})(n[k[N_1] \mid H_1] \mid N_2 \mid H_2)$. Since for all processes P, $M' \bullet P \; \mathcal{S} \; N' \bullet P$, it also holds $M' \bullet H_1 \; \mathcal{S} \; N' \bullet H_1$, and $O_1 \; \mathcal{S} \; O_2$ follows because $\mathcal{S}$ is closed under parallel composition and restriction.

∗ $k \in \tilde{m}$. As $D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{m})\langle k[M_1]\rangle M_2$, it also holds $D[M] \xrightarrow{\texttt{enter\_}n} M' \equiv (\boldsymbol{\nu}\tilde{m})(n[k[M_1] \mid \circ] \mid M_2)$. The induction hypothesis tells us that there exists a system N' such that $D[N] \mid n[\,\circ\,] \Rightarrow N' \equiv (\boldsymbol{\nu}\tilde{n})(n[N_1 \mid \circ] \mid N_2)$, and for all processes P, it holds $M' \bullet P \; \mathcal{S} \; N' \bullet P$. We can derive $D[N] \mid H \Rightarrow O_2 \equiv (\boldsymbol{\nu}\tilde{h}, \tilde{n})(n[N_1 \mid H_1] \mid N_2 \mid H_2)$. Since for all processes P, $M' \bullet P \; \mathcal{S} \; N' \bullet P$, it also holds $M' \bullet H_1 \; \mathcal{S} \; N' \bullet H_1$, and $O_1 \; \mathcal{S} \; O_2$ follows because $\mathcal{S}$ is closed under parallel composition and restriction.

– $D[M] \mid H \xrightarrow{\tau} O_1$, because

$$D[M] \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{m})\langle M_1\rangle M_2 \text{ and } H \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{h})\langle k[H_1]\rangle H_2.$$

Then $O_1 \equiv (\boldsymbol{\nu}\tilde{h}, \tilde{m})(n[k[H_1] \mid M_1] \mid M_2 \mid H_2)$. As $D[M] \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{m})\langle M_1\rangle M_2$, it also holds $D[M] \xrightarrow{n.\overline{\texttt{enter}}\_k} M' \equiv (\boldsymbol{\nu}\tilde{m})(n[k[\,\circ\,] \mid M_1] \mid M_2)$. The induction hypothesis tells us that there exists a system N' such that $D[N] \stackrel{n.\overline{\texttt{enter}}\_k}{\Longrightarrow} N' \equiv (\boldsymbol{\nu}\tilde{n})(n[k[\,\circ\,] \mid N_1] \mid N_2)$, and for all processes P, it holds $M' \bullet P \; \mathcal{S} \; N' \bullet P$. As $D[N] \stackrel{n.\overline{\texttt{enter}}\_k}{\Longrightarrow} N'$, we can derive $D[N] \stackrel{\texttt{amb\_}k}{\Longrightarrow} (\boldsymbol{\nu}\tilde{n})\langle N_1\rangle N_2$. It follows $D[N] \mid H \Rightarrow (\boldsymbol{\nu}\tilde{h}, \tilde{n})(n[k[H_1] \mid N_1] \mid N_2 \mid H_2) = O_2$. Since for all processes P, it holds $M' \bullet P \; \mathcal{S} \; N' \bullet P$, we have $M' \bullet h[H_1] \; \mathcal{S} \; N' \bullet h[H_1]$, and $O_1 \; \mathcal{S} \; O_2$ follows because $\mathcal{S}$ is closed under parallel composition and restriction.

• $C[-] = n[D[-] \mid P]$, where P is an arbitrary process.

We know that $D[M] \; \mathcal{S} \; D[N]$ satisfies the bisimulation conditions in $\mathcal{S}$, and we want to prove that $n[D[M] \mid P] \; \mathcal{S} \; n[D[N] \mid P]$ behaves as a bisimulation as well. We perform a case analysis on the transition $n[D[M] \mid P] \xrightarrow{\alpha} O_1$.

– $n[D[M] \mid P] \xrightarrow{\tau} O_1$, because $D[M] \xrightarrow{\tau} M'$. Then $O_1 \equiv n[M' \mid P]$. The induction hypothesis tells us that there exists a system N' such that $D[N] \Rightarrow N'$ and $M' \mathcal{S} N'$. We can derive $n[D[N] \mid P] \Rightarrow n[N' \mid P]$ and conclude $n[M' \mid P] \mathcal{S} n[N' \mid P]$ because $\mathcal{S}$ is closed under ambient.

– $n[D[M] \mid P] \xrightarrow{\tau} O_1$, because $P \xrightarrow{\tau} P'$. Then $O_1 \equiv n[D[M] \mid P']$. Call $O_2 = n[D[N] \mid P']$. Then $O_1 \mathcal{S} O_2$ because $D[M] \mathcal{S} D[N]$, and $\mathcal{S}$ is closed under the contexts of the form $C[-] = n[- \mid Q]$ where Q is a process.

– $n[D[M] \mid P] \xrightarrow{\tau} O_1$, because $D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$. Then $O_1 \equiv (\boldsymbol{\nu}\tilde{r})(k[M_1] \mid n[M_2 \mid P])$. We distinguish the two cases $k \in \tilde{r}$ and $k \notin \tilde{r}$.

  * $k \notin \tilde{r}$. From $D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ we can derive $D[M] \xrightarrow{k.\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})(k[M_1] \mid n[\circ \mid M_2])$. The induction hypothesis tells us that there exists a system N' such that $D[N] \stackrel{k.\texttt{exit\_}n}{\Longrightarrow} N' \equiv (\boldsymbol{\nu}\tilde{s})(k[N_1] \mid n[\circ \mid N_2])$ and for all processes Q, it holds $M' \bullet Q \mathcal{S} N' \bullet Q$. But $D[N] \stackrel{k.\texttt{exit\_}n}{\Longrightarrow} N'$ can only be derived from $D[N] \stackrel{\texttt{exit\_}n}{\Longrightarrow} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$ and thus $n[D[N] \mid P] \Rightarrow N' \bullet P$. As for all processes Q, it holds $M' \bullet Q \mathcal{S} N' \bullet Q$, we can derive $(\boldsymbol{\nu}\tilde{r})(k[M_1] \mid n[P \mid M_2]) \mathcal{S} (\boldsymbol{\nu}\tilde{s})(k[N_1] \mid n[P \mid N_2])$, as required.
  * $k \in \tilde{r}$. From $D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ we can derive $D[M] \xrightarrow{*.\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})(k[M_1] \mid n[\circ \mid M_2])$. The induction hypothesis tells us that there exists a system N' such that $n[\circ \mid D[N]] \Rightarrow N' \equiv (\boldsymbol{\nu}\tilde{s})(k[N_1] \mid n[\circ \mid N_2])$, and for all processes Q, it holds $M' \bullet Q \mathcal{S} N' \bullet Q$. We can instantiate the placeholder $\circ$ with the process P, thus obtaining the transition $n[D[N] \mid P] \Rightarrow N' \bullet P$. As for all processes Q, it holds $M' \bullet Q \mathcal{S} N' \bullet Q$, we have $O_1 = (\boldsymbol{\nu}\tilde{m})(k[M_1] \mid n[P \mid M_2]) \equiv M' \bullet P \mathcal{S} N' \bullet P \equiv (\boldsymbol{\nu}\tilde{s})(k[N_1] \mid n[P \mid N_2]) = O_2$, as required.

– $n[D[M] \mid P] \xrightarrow{\tau} O_1$, because $P \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[P_1]\rangle P_2$. This implies $O_1 \equiv (\boldsymbol{\nu}\tilde{r})(k[P_1] \mid n[D[M] \mid P_2])$. It also holds $n[D[N] \mid P] \xrightarrow{\tau} \equiv (\boldsymbol{\nu}\tilde{r})(k[P_1] \mid n[D[N] \mid P_2])$. Call this last term O_2. The relation $O_1 \mathcal{S} O_2$ follows because $D[M] \mathcal{S} D[N]$ and from the closure properties of $\mathcal{S}$.

– $n[D[M] \mid P] \xrightarrow{\tau} O_1$, and the τ action is generated by an interaction between $D[M]$ and P. There are three cases.

  * $D[M] \xrightarrow{\texttt{amb\_}m} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2$ and $P \xrightarrow{\texttt{open\_}m} P'$. Then $O_1 = n[(\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2) \mid P']$. It holds $D[M] \xrightarrow{n.\texttt{open\_}m} n[\circ \mid (\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2)]$. The induction hypothesis tells us that there exists a system N' such that $D[N] \xrightarrow{n.\texttt{open\_}m} N'$, and for all processes Q it holds $M' \bullet Q \mathcal{S} N' \bullet Q$. The system N' must be of the form $n[\circ \mid (\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2)]$. The transition $D[N] \stackrel{n.\texttt{open\_}m}{\Longrightarrow} N'$ must have been derived from $D[N] \xrightarrow{\texttt{amb\_}m} (\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2$. This implies that $n[D[N] \mid P] \Rightarrow n[(\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2) \mid P']$. Call this last term O_2. We can instantiate the placeholder $\circ$ with the process P', thus obtaining the transition $n[D[N] \mid P] \Rightarrow N' \bullet P$. As for all processes Q, it holds $M' \bullet Q \mathcal{S} N' \bullet Q$, we have $O_1 = n[(\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2) \mid P'] \equiv M' \bullet P' \mathcal{S} N' \bullet P' \equiv n[(\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2) \mid P'] = O_2$, as required.
  * $D[M] \xrightarrow{\texttt{enter\_}m}$ and $P \xrightarrow{\texttt{amb\_}m}$, or $D[M] \xrightarrow{\texttt{amb\_}m}$ and $P \xrightarrow{\texttt{enter\_}m}$. Call A_1 the outcome of the interaction between $D[M]$ and P. In both cases, by an analysis carried on previously, we know that there is a process A_2 such that

$D[N] \mid P \Rightarrow A_2$, with $A_1 \mathcal{S} A_2$. We obtain $n[D[M] \mid P] \xrightarrow{\tau} n[A_1] = O_1$, and $n[D[N] \mid P] \Rightarrow n[A_2]$. The relation $n[A_1] \mathcal{S} n[A_2]$ follows from the closure of $\mathcal{S}$ under ambient.

- $n[D[M] \mid P] \xrightarrow{n.\overline{\texttt{enter}}\_k} O_1$. Then $O_1 \equiv n[k[\,\circ\,] \mid D[M] \mid P]$. But $n[D[N] \mid P] \xrightarrow{n.\overline{\texttt{enter}}\_k} O_2$, where $O_2 \equiv n[k[\circ] \mid D[N] \mid P]$. For all processes Q, $O_1 \bullet Q \mathcal{S} O_2 \bullet Q$ follows from $D[M] \mathcal{S} D[N]$ because of the closure properties of $\mathcal{S}$.
- $n[D[M] \mid P] \xrightarrow{n.\texttt{exit}\_m} m[\,\circ\,] \mid n[D[M] \mid P'] = O_1$, because $P \xrightarrow{\texttt{out}\_m} P'$. It also holds $n[D[N] \mid P] \xrightarrow{n.\texttt{exit}\_m} m[\,\circ\,] \mid n[D[N] \mid P']$. Call this last term O_2. Then, for all processes Q, the relation $O_1 \bullet Q \mathcal{S} O_2 \bullet Q$ follows from $D[M] \mathcal{S} D[N]$ because of the closure properties of $\mathcal{S}$.

□

Proof of Lemma 3.9 – omitted cases

Case $\alpha = k.\texttt{exit}\_n$**.** Let P be a process. We know that $M \xrightarrow{k.\texttt{exit}\_n} M'$. Then

$$M \equiv (\boldsymbol{\nu}\tilde{m})(k[\texttt{out}\_n.M_1 \mid M_2] \mid M_3)$$

where $(\{n,k\} \cup \mathrm{fn}(P)) \cap \{\tilde{m}\} = \emptyset$, and

$$M' \equiv (\boldsymbol{\nu}\tilde{m})(k[M_1 \mid M_2] \mid n[\circ \mid M_3]).$$

Now,

$$\begin{array}{rl}
 & C_{k.\texttt{exit}_n} M \bullet P \\
\equiv & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)a[\texttt{in}_k.\texttt{out}_k.\mathsf{done}[\texttt{out}_a]] \mid n[P \mid k[\texttt{out}_n.M_1 \mid M_2] \mid M_3)] \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)a[\texttt{in}_k.\texttt{out}_k.\mathsf{done}[\texttt{out}_a]] \mid k[M_1 \mid M_2] \mid n[P \mid M_3]) \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)k[a[\texttt{out}_k.\mathsf{done}[\texttt{out}_a]] \mid M_1 \mid M_2] \mid n[P \mid M_3]) \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)a[\mathsf{done}[\texttt{out}_a]] \mid k[M_1 \mid M_2] \mid n[P \mid M_3]) \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)(\mathsf{done}[\,] \mid a[\,]) \mid k[M_1 \mid M_2] \mid n[P \mid M_3]) \\
\cong_{\mathrm{s}} & (\boldsymbol{\nu}\tilde{m})(k[M_1 \mid M_2] \mid n[\circ \mid P_3]) \bullet P \mid \mathsf{done}[\,] \\
= & M' \bullet P \mid \mathsf{done}[\,]
\end{array}$$

This implies $C_{k.\texttt{exit}\_n}[M] \bullet P \Rightarrow \cong_{\mathrm{s}} M' \bullet P \mid \mathsf{done}[\,]$.

Case $\alpha = n.\overline{\texttt{enter}}\_k$**.** Let P be a process. We know that $M \xrightarrow{n.\overline{\texttt{enter}}\_k} M'$. Then

$$M \equiv (\boldsymbol{\nu}\tilde{m})(n[M_1] \mid M_2)$$

where $(\{n,k\} \cup \mathrm{fn}(P)) \cap \{\tilde{m}\} = \emptyset$, and

$$M' \equiv (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid k[\,\circ\,]] \mid M_2).$$

Now,

$$\begin{array}{ll}
& C_{n.\overline{\texttt{enter}}_k}[M] \bullet P \\
\equiv & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)a[\texttt{in}_n.k[\texttt{out}_a.(P \mid (\boldsymbol{\nu}b)b[\texttt{out}_k.\texttt{out}_n.\mathsf{done}[\texttt{out}_b]])]] \mid n[M_1] \mid M_2) \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[k[\texttt{out}_a.(P \mid (\boldsymbol{\nu}b)b[\texttt{out}_k.\texttt{out}_n.\mathsf{done}[\texttt{out}_b]])]]] \mid M_2) \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[] \mid k[P \mid (\boldsymbol{\nu}b)b[\texttt{out}_k.\texttt{out}_n.\mathsf{done}[\texttt{out}_b]]]] \mid M_2) \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[] \mid k[P] \mid (\boldsymbol{\nu}b)b[\texttt{out}_n.\mathsf{done}[\texttt{out}_b]]] \mid M_2) \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[] \mid k[P]] \mid (\boldsymbol{\nu}b)b[\mathsf{done}[\texttt{out}_b]] \mid M_2) \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[] \mid k[P]] \mid (\boldsymbol{\nu}b)b[] \mid \mathsf{done}[] \mid M_2) \\
\cong_{\mathrm{s}} & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid k[\circ]] \mid M_2) \bullet P \mid \mathsf{done}[] \\
= & M' \bullet P \mid \mathsf{done}[]
\end{array}$$

This implies $C_{n.\overline{\texttt{enter}}\_k}[M] \bullet P \Rightarrow\!\cong_{\mathrm{s}} M' \bullet P \mid \mathsf{done}[]$.

Case $\alpha = k.\texttt{open}\_n$**.** Let P be a process. We know that $M \xrightarrow{k.\texttt{open}\_n} M'$. Then $M \equiv (\boldsymbol{\nu}\tilde{m})(n[M_1] \mid M_2)$, where $n \in \{\tilde{m}\}$, and $M' \equiv k[\circ \mid (\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2)]$. Names a and b are fresh for M. Now,

$$\begin{array}{ll}
& C_{k.\texttt{open}_n}[M] \bullet P \\
\equiv & k[P \mid (\boldsymbol{\nu}a,b)(\texttt{open}_b.\texttt{open}_a.\mathsf{done}[\texttt{out}_k] \mid \\
& \qquad a[(\boldsymbol{\nu}\tilde{m})(n[M_1] \mid M_2) \mid \texttt{open}_n.b[\texttt{out}_a]])] \\
\xrightarrow{\tau} & k[P \mid (\boldsymbol{\nu}a,b)(\texttt{open}_b.\texttt{open}_a.\mathsf{done}[\texttt{out}_k] \mid a[(\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2) \mid b[\texttt{out}_a]])] \\
\xrightarrow{\tau} & k[P \mid (\boldsymbol{\nu}a,b)(\texttt{open}_b.\texttt{open}_a.\mathsf{done}[\texttt{out}_k] \mid a[(\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2)] \mid b[])] \\
\xrightarrow{\tau} & k[P \mid (\boldsymbol{\nu}a,b)(\texttt{open}_a.\mathsf{done}[\texttt{out}_k] \mid a[(\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2)])] \\
\xrightarrow{\tau} & k[P \mid (\boldsymbol{\nu}a,b)(\mathsf{done}[\texttt{out}_k] \mid (\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2))] \\
\xrightarrow{\tau} & k[P \mid (\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2))]\mathsf{done}[] \\
\equiv & k[\circ \mid (\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2)] \bullet P \mid \mathsf{done}[] \\
= & M' \bullet P \mid \mathsf{done}[]
\end{array}$$

This implies $C_{k.\texttt{open}\_n}[M] \bullet P \Rightarrow\!\cong_{\mathrm{s}} M' \bullet P \mid \mathsf{done}[]$. □

Proof of Lemma 3.11

Part 1. For point 1), the definition of $\bullet$ assures that there exists an arbitrary context $C[-]$ such that $C[\mathsf{spy}_\alpha\langle i,j,P\rangle] = M \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$, and names in P are not bound in $C[-]$. The construction of $\mathsf{spy}_\alpha\langle i,j,P\rangle$ assures that if $C[\mathsf{spy}_\alpha\langle i,j,P\rangle] \xrightarrow{\tau} Q$, then either there is an arbitrary context C' such that $Q = C'[\mathsf{spy}_\alpha\langle i,j,P\rangle]$, or $Q = C[P']$ where $\mathsf{spy}_\alpha\langle i,j,P\rangle \xrightarrow{\tau} P'$. But if $\mathsf{spy}_\alpha\langle i,j,P\rangle \xrightarrow{\tau} P'$, then $P' \Downarrow i \not\Downarrow j$, or $P' \Downarrow j \not\Downarrow i$. As $O \Downarrow i,j$, O must be the outcome of the first reduction, and as such there exists an arbitrary context $C'[-]$ such that $O = C'[\mathsf{spy}_\alpha\langle i,j,P\rangle]$. Let $M' = C'[\circ]$. As $C[\mathsf{spy}_\alpha\langle i,j,P\rangle] \xrightarrow{\tau} C'[\mathsf{spy}_\alpha\langle i,j,P\rangle]$, names in P cannot be bound in $C'[-]$. This implies $O = C'[\mathsf{spy}_\alpha\langle i,j,P\rangle] = M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$, as required for 1).

For point 2), $M \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle = C[\mathsf{spy}_\alpha\langle i,j,P\rangle] \xrightarrow{\tau} C'[\mathsf{spy}_\alpha\langle i,j,P\rangle] = M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$ implies $M = C[\circ] \xrightarrow{\tau} C'[\circ] = M'$, as required.

Part 2. It is easy to see that the relation

$$\mathcal{R} = \{(n[(\boldsymbol{\nu} i, j)\mathsf{spy}_\alpha\langle i, j, P\rangle \mid R], n[P \mid R]) \mid \text{for all } n, R\} \ \cup \mathcal{I}$$

is a bisimulation up-to context. Observe that the soundness of the up-to context proof technique does not depend on the completeness of the bisimilarity. □

Proof of Lemma 3.12 The relation $\{((\boldsymbol{\nu} n)n[\,], \mathbf{0})\}$ is a bisimulation, and the result follows from the soundness of bisimulation. □

Proof of Lemma 3.14 – omitted cases

Case $\alpha = k.\mathtt{enter\_}n$**.** Observe that

$$C_\alpha[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle = n[\mathsf{done}[\mathtt{in\_}k.\mathtt{out\_}k.\mathtt{out\_}n] \mid \mathsf{spy}_\alpha\langle i, j, P\rangle] \mid M \ .$$

As $N \Downarrow i, j$ and done is fresh, by Lemma 3.11(1), there must be a system $D[-]$ such that $N \mid \mathsf{done}[\,] \equiv D[\mathsf{done}[\,]] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle$ and $C_\alpha[M] \Rightarrow D[\mathsf{done}[\,]]$. As P cannot reduce and done is fresh, the ambient n does not migrate during the reduction. Moreover, as M is a system, the ambient n cannot be opened. Also observe that the ambient done must consume the prefix $\mathtt{in\_}k$, thus requiring the presence of an ambient k inside the ambient n during the reduction. More precisely, there exist systems M_1 and M_2 and a static context $C[-]$ such that:

$$\begin{array}{rl}
 & C_\alpha[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \\
= & n[\mathsf{done}[\mathtt{in_}k.\mathtt{out_}k.\mathtt{out_}n] \mid \mathsf{spy}_\alpha\langle i, j, P\rangle] \mid M \\
\Rightarrow \xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[\mathsf{done}[\mathtt{in_}k.\mathtt{out_}k.\mathtt{out_}n] \mid \mathsf{spy}_\alpha\langle i, j, P\rangle \mid M_1] \mid M_2) \\
\xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[\mathsf{spy}_\alpha\langle i, j, P\rangle \mid C[\mathsf{done}[\mathtt{out_}k.\mathtt{out_}n]]] \mid M_2) \\
\Rightarrow & D[\mathsf{done}[\,]] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \\
\equiv & D[\mathbf{0}] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \mid \mathsf{done}[\,] \\
\equiv & N \mid \mathsf{done}[\,]
\end{array}$$

Examining the above reductions sequence from $C_\alpha[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle$ we conclude that

$$M \ \Rightarrow \xrightarrow{k.\mathtt{enter\_}n} (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid \circ] \mid M_2).$$

As the name done is fresh for M, by Lemma 3.13 we also have that

$$(\boldsymbol{\nu}\tilde{m})(n[\circ \mid \mathbf{0} \mid M_1] \mid M_2) \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \ \Rightarrow D[\mathbf{0}] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle.$$

Repeated application of Lemma 3.11(2) gives $(\boldsymbol{\nu}\tilde{m})(n[\circ \mid \mathbf{0} \mid M_1] \mid M_2) \ \Rightarrow D[\mathbf{0}]$, and therefore, as $\equiv$ is closed under reduction, there is a M', $M' \equiv D[\mathbf{0}]$, such that $M \xRightarrow{k.\mathtt{enter\_}n} M'$, as desired.

Case $\alpha = k.\mathtt{exit\_}n$**.** Observe that

$$C_{k.\mathtt{exit\_}n}[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \equiv (\boldsymbol{\nu} a)a[\mathtt{in\_}k.\mathtt{out\_}k.\mathsf{done}[\mathtt{out\_}a]] \mid n[\mathsf{spy}_\alpha\langle i, j, P\rangle \mid M] \ .$$

To unleash the ambient done, the ambient a must perform both its capabilities, and as its name is restricted the ambient a will be empty at the end of reduction. As P cannot reduce, and M is a system, the ambient n does not migrate during the reduction. Also observe that the ambient

a must consume the prefix $\mathtt{in}\_k$, thus requiring the presence of an ambient k at top-level. More precisely, there exist a system M_1 and static contexts $D[-]$ and $E[-_1, -_2]$ such that:

$$\begin{array}{rl} & C_{k.\mathtt{exit}\_n}[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \\ = & (\boldsymbol{\nu} a)a[\mathtt{in}\_k.\mathtt{out}\_k.\mathsf{done}[\mathtt{out}\_a]] \mid n[\mathsf{spy}_\alpha\langle i, j, P\rangle \mid M] \\ \Rightarrow & (\boldsymbol{\nu} a)a[\mathtt{in}\_k.\mathtt{out}\_k.\mathsf{done}[\mathtt{out}\_a]] \mid M_1 \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \\ \xrightarrow{\tau} & (\boldsymbol{\nu} a)D[a[\mathtt{out}\_k.\mathsf{done}[\mathtt{out}\_a]]] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \\ \Rightarrow & (\boldsymbol{\nu} a)E[\mathsf{done}[\,], a[\,]] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \qquad (\star) \\ \equiv & N \mid \mathsf{done}[\,] \end{array}$$

Examining the above reductions sequence from $C_{k.\mathtt{exit}\_n}[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle$ we conclude that

$$M \;\Rightarrow \xrightarrow{k.\mathtt{exit}\_n} M_1.$$

As the name done is fresh for M, by several applications of Lemma 3.13 to the reduction marked by $(\star)$ we have:

$$(\boldsymbol{\nu} a)a[\mathtt{in}\_k.\mathtt{out}\_k.\mathbf{0}] \mid M_1 \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \;\Rightarrow (\boldsymbol{\nu} a)E[\mathbf{0}, a[\,]] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \;.$$

Again, as a is fresh, by several applications of Lemma 3.13, and reducing underneath $(\boldsymbol{\nu} a)$, we obtain:

$$(\boldsymbol{\nu} a)(\mathbf{0} \mid M_1) \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \;\Rightarrow (\boldsymbol{\nu} a)E[\mathbf{0}, \mathbf{0}] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \;.$$

Summarising,

$$M_1 \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \equiv (\boldsymbol{\nu} a)(\mathbf{0} \mid M_1) \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \;\Rightarrow (\boldsymbol{\nu} a)E[\mathbf{0}, \mathbf{0}] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle$$

and, as $\equiv$ is closed under reductions,

$$M_1 \;\Rightarrow\!\equiv E[\mathbf{0}, \mathbf{0}] \;.$$

So, assuming $M' = E[\mathbf{0}, \mathbf{0}]$, we can conclude.

Case $\alpha = k.\mathtt{open}\_n$**.** Observe that

$$\begin{array}{l} C_{k.\mathtt{open}\_n}[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle = \\ \quad k[\mathsf{spy}_\alpha\langle i, j, P\rangle \mid (\boldsymbol{\nu} a, b)(\mathtt{open}\_b.\mathtt{open}\_a.\mathsf{done}[\mathtt{out}\_k] \mid a[M \mid \mathtt{open}\_n.b[\mathtt{out}\_a]])] \end{array}$$

where a and b are fresh. To unleash the ambient done, the ambient a must use its $\mathtt{open}\_n$ capability, and the ambient b must exit from a. Moreover both the empty ambients a and b will be opened before done is activated. Also observe that the prefix $\mathtt{open}\_n$ must be consumed, thus requiring the presence of an ambient n inside the ambient a. More precisely, there exist a

system M_1, processes Q_i, and a static context $D[-]$ such that:

$$\begin{array}{rl}
& C_{k.\mathtt{open}_n}[M] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \\
= & k[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid a[M \mid \mathtt{open}_n.b[\mathtt{out}_a]])] \\
\Rightarrow & k[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid a[M_1 \mid \mathtt{open}_n.b[\mathtt{out}_a]])] \\
\stackrel{\tau}{\longrightarrow} & k[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid a[Q \mid b[\mathtt{out}_a]])] \\
\Rightarrow & k[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid a[Q_1 \mid b[\mathtt{out}_a]])] \\
\stackrel{\tau}{\longrightarrow} & k[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid b[\,] \mid a[Q_1])] \\
\Rightarrow & k[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid b[\,] \mid a[Q_2])] \\
\stackrel{\tau}{\longrightarrow} & k[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid \mathbf{0} \mid a[Q_2])] \\
\Rightarrow & k[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid \mathbf{0} \mid a[Q_3])] \\
\Rightarrow & k[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathsf{done}[\mathtt{out}_k] \mid \mathbf{0} \mid Q_3)] \\
\Rightarrow & D[\mathsf{done}[\,]] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \\
\equiv & D[\mathbf{0}] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \mid \mathsf{done}[\,] \\
= & N \mid \mathsf{done}[\,]
\end{array}$$

Examining the above reductions sequence from $C_{k.\mathtt{open}\_n}[M] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$ we conclude that

$$M \;\Rightarrow \xrightarrow{k.\mathtt{open}\_n} k[\circ \mid Q].$$

As

$$\begin{array}{l}
k[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid a[Q \mid b[\mathtt{out}_a]])] \\
\Rightarrow D[\mathsf{done}[\,]] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle
\end{array}$$

and the name done is fresh, by several applications of Lemma 3.13 we have

$$\begin{array}{l}
k[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathbf{0} \mid a[Q \mid b[\mathtt{out}_a]])] \\
\Rightarrow D[\mathbf{0}] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle.
\end{array}$$

By Lemma 3.11, this implies

$$k[\circ \mid (\boldsymbol{\nu} a,b)(\mathtt{open}\_b.\mathtt{open}\_a.\mathbf{0} \mid a[Q \mid b[\mathtt{out}\_a]])] \;\Rightarrow D[\mathbf{0}].$$

Applying our proof techniques we can easily prove that:

$$k[\circ \mid (\boldsymbol{\nu} a,b)(\mathtt{open}\_b.\mathtt{open}\_a.\mathbf{0} \mid a[Q \mid b[\mathtt{out}\_a]])] \cong_{\mathrm{s}} k[\circ \mid Q].$$

As $\cong_{\mathrm{s}}$ is closed under reduction, it follows that there is M' such that

$$k[\circ \mid Q] \;\Rightarrow M' \cong_{\mathrm{s}} D[\mathbf{0}].$$

So, there is M' such that $M \;\Rightarrow M'$ and $N \cong_{\mathrm{s}} M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$, as desired. □

C Proofs from Section 5

Proof of Lemma 5.4

Let $\mathcal{R} = \{(P', Q') : P' \equiv (\boldsymbol{\nu} n)P,\ Q' \equiv (\boldsymbol{\nu} n)Q,\ P \cong^{\mathrm{e}}_{\mathrm{p}} Q\} \cup \cong^{\mathrm{e}}_{\mathrm{p}}$. We show that $\mathcal{R} \subseteq \cong^{\mathrm{e}}_{\mathrm{p}}$. The relation $\mathcal{R}$ is reduction closed because both $\cong^{\mathrm{e}}_{\mathrm{p}}$ and $\equiv$ are, and restriction does not influence internal reductions. $\mathcal{R}$ is also barb preserving because $\cong^{\mathrm{e}}_{\mathrm{p}}$ and $\equiv$ are. To prove that $\mathcal{R}$ is closed under ambient nesting, we have to show that if $P' \mathcal{R} Q'$, with $P' \equiv (\boldsymbol{\nu} n)P$ and $Q' \equiv (\boldsymbol{\nu} n)Q$, then $k[P'] \mathcal{R} k[Q']$. But $k[P'] \equiv k[(\boldsymbol{\nu} n)P] \equiv (\boldsymbol{\nu} n)k[P]$ and $k[Q'] \equiv k[(\boldsymbol{\nu} n)Q] \equiv (\boldsymbol{\nu} n)k[Q]$. Moreover, by definition of $\cong^{\mathrm{e}}_{\mathrm{p}}$, $k[P] \cong^{\mathrm{e}}_{\mathrm{p}} k[Q]$. The result follows from the construction of $\mathcal{R}$. The argument for parallel composition is similar. □

Proof of Lemma 5.5

To prove the inclusion $\cong^{\mathrm{e}}_{\mathrm{p}} \cap (\mathcal{M} \times \mathcal{M}) \subseteq \cong_{\mathrm{s}}$, observe that the relation $\cong^{\mathrm{e}}_{\mathrm{p}} \cap (\mathcal{M} \times \mathcal{M})$ is: reduction closed because $\cong^{\mathrm{e}}_{\mathrm{p}}$ is reduction closed and systems always reduce in systems; barb preserving because $\cong^{\mathrm{e}}_{\mathrm{p}}$ preserves barbs; closed under system contexts because $\cong^{\mathrm{e}}_{\mathrm{p}}$ is preserved by parallel composition, ambient, and, by Lemma 5.4, by restriction.

□

Proof of Theorem 5.2 – omitted cases

We prove that the relation $\cong^{\mathrm{e}}_{\mathrm{p}}$ is preserved by prefixing. We have to prove that if $P \cong^{\mathrm{e}}_{\mathrm{p}} Q$, then $\pi.P \cong^{\mathrm{e}}_{\mathrm{p}} \pi.Q$. Rather than working directly with $\cong^{\mathrm{e}}_{\mathrm{p}}$, we use Theorem 5.3 and we prove that $\pi.P \mathcal{S} \pi.Q$. For that, we must show that for all n, R, it holds $n[\pi.P \mid R] \approx n[\pi.Q \mid R]$. We perform a case analysis on π.

$\pi = \mathtt{in}\_o$. We show that the relation

$$\mathcal{R} = \{(n[\mathtt{in}\_o.P \mid R], n[\mathtt{in}\_o.Q \mid R]) : P \cong^{\mathrm{e}}_{\mathrm{p}} Q,\ n, R \text{ arbitrary}\}^{=} \cup \approx$$

is a bisimulation up to context and up to structural congruence. Suppose $n[\mathtt{in}\_o.P \mid R] \xrightarrow{\alpha} M$. We perform a case analysis on α.

$\alpha = \tau$. There are two sub-cases.
First case. $M \equiv n[\mathtt{in}\_o.P \mid R']$ with $R \xrightarrow{\tau} R'$. It follows that $n[\mathtt{in}\_o.Q \mid R] \xrightarrow{\tau} N$, where $N \equiv n[\mathtt{in}\_o.Q \mid R']$, and $M \equiv \mathcal{R} \equiv N$.
Second case. $M \equiv (\boldsymbol{\nu} \tilde{r}_2)(r[R_1'] \mid n[\mathtt{in}\_o.P \mid R'])$, where $R \equiv (\boldsymbol{\nu} \tilde{r})(r[R_1] \mid R_2)$ and $R' \equiv (\boldsymbol{\nu} \tilde{r}_1)R_2$, with $\tilde{r} = \tilde{r}_1 \cup \tilde{r}_2$ and $r \notin \tilde{r}$. This implies $n[\mathtt{in}\_o.Q \mid R] \xrightarrow{\tau} N$, where $N \equiv (\boldsymbol{\nu} \tilde{r}_1)(r[R_1] \mid n[\mathtt{in}\_o.Q \mid R_2])$. Now, we can factor out the system context $C[-] = (\boldsymbol{\nu} \tilde{r}_1)(r[R_1] \mid -)$ and the construction of $\mathcal{R}$ guarantees that we are still in $\mathcal{R}$ up to context and up to $\equiv$.

$\alpha = m.\mathtt{open}\_n$. Then $M \equiv n[\circ \mid \mathtt{in}\_o.P \mid R]$. This implies $n[\mathtt{in}\_o.Q \mid R] \xrightarrow{m.\mathtt{open}\_n} N$, where $N \equiv n[\circ \mid \mathtt{in}\_o.Q \mid R]$. Then, for all processes R' we have $M \bullet R' \equiv \mathcal{R} \equiv N \bullet R'$.

$\alpha = n.\overline{\mathtt{enter}}\_k$. Then $M \equiv n[\mathtt{in}\_o.P \mid R \mid k[\,\circ\,]]$. This implies $n[\mathtt{in}\_o.Q \mid R] \xrightarrow{n.\overline{\mathtt{enter}}\_k} N$, where $N \equiv n[\mathtt{in}\_o.Q \mid R \mid k[\,\circ\,]]$. Then for all processes R' we have $M \bullet R' \equiv \mathcal{R} \equiv N \bullet R'$.

$\alpha = n.\mathtt{exit}\_k$. Then $M \equiv n[\mathtt{in}\_o.P \mid R'] \mid k[\circ]$ and R has unleashed the capability $\mathtt{out}\_k$ turning into R'. This implies $n[\mathtt{in}\_o.Q \mid R] \xrightarrow{n.\mathtt{exit}\_k} N$, where $N \equiv n[\mathtt{in}\_o.Q \mid R'] \mid k[\,\circ\,]$. Then, factoring out the context $C[-] = - \mid k[S]$, for all processes S, the construction of $\mathcal{R}$ guarantees that we are still in $\mathcal{R}$ up to context and up to $\equiv$.

$\alpha = n.\texttt{enter\_}o$. There are two sub-cases.

First case. $M \equiv o[n[\texttt{in\_}o.P \mid R'] \mid \circ]$ and R has unleashed the capability $\texttt{in\_}o$ turning into R'. This implies $n[\texttt{in\_}o.Q \mid R] \xrightarrow{n.\texttt{enter\_}o} N$, where $N \equiv o[n[\texttt{in\_}o.Q \mid R'] \mid \circ]$. Then, factoring out the context $C[-] = o[- \mid S]$, for all processes S, the construction of $\mathcal{R}$ guarantees that we are still in $\mathcal{R}$ up to context and up to $\equiv$.

Second case. $M \equiv o[n[P \mid R] \mid \circ]$. This implies $n[\texttt{in\_}o.Q \mid R] \xrightarrow{n.\texttt{enter\_}o} N$, where $N \equiv o[n[Q \mid R] \mid \circ]$. As $P \cong^{\mathrm{e}}_{\mathrm{p}} Q$ it holds that $n[P \mid R] \cong^{\mathrm{e}}_{\mathrm{p}} n[Q \mid R]$. By Theorem 5.3 we get $M \bullet S \equiv\approx\equiv N \bullet S$ and hence $M \bullet S \;\mathcal{R}\; N \bullet S$.

$\alpha = n.\texttt{enter\_}k$, $k \neq o$. It is similar to the first part of the previous case.

$\pi = \texttt{out\_}o$. We show that the relation

$$\mathcal{R} = \{(n[\texttt{out\_}o.P \mid R], n[\texttt{out\_}o.Q \mid R]) : \; P \cong^{\mathrm{e}}_{\mathrm{p}} Q\}^{=} \cup \; \approx$$

is a bisimulation up to context and up to structural congruence. The only case different from the above is when the process $\texttt{out\_}o.P$ exercises the capability $\texttt{out\_}o$. Suppose $n[\texttt{out\_}o.P \mid R] \xrightarrow{n.\texttt{exit\_}o} M \equiv n[P \mid R] \mid o[\,\circ\,]$. This implies $n[\texttt{out\_}o.Q \mid R] \xrightarrow{n.\texttt{exit\_}o} N \equiv n[Q \mid R] \mid o[\,\circ\,]$. As $P \cong^{\mathrm{e}}_{\mathrm{p}} Q$ it holds that $n[P \mid R] \cong^{\mathrm{e}}_{\mathrm{p}} n[Q \mid R]$. By Lemma 5.5 and Theorem 3.17 it follows $n[P \mid R] \approx n[Q \mid R]$. As $\approx$ is preserved by system contexts, we have $M \bullet S \equiv\approx\equiv N \bullet S$. As a consequence, $M \bullet S \;\mathcal{R}\; N \bullet S$.

$\pi = \texttt{open\_}o$. We show that the relation

$$\mathcal{R} = \{(n[\texttt{open\_}o.P \mid R], n[\texttt{open\_}o.Q \mid R]) : \; P \cong^{\mathrm{e}}_{\mathrm{p}} Q\}^{=} \cup \; \approx$$

is a bisimulation up to context and up to structural congruence. The only case different from the above is when the process $\texttt{open\_}o.P$ exercises the capability $\texttt{open\_}o$. Suppose $n[\texttt{open\_}o.P \mid R] \xrightarrow{\tau} n[P \mid R']$. This implies $n[\texttt{open\_}o.Q \mid R] \xrightarrow{\tau} n[Q \mid R']$. As $P \cong^{\mathrm{e}}_{\mathrm{p}} Q$ it holds that $n[P \mid R'] \cong^{\mathrm{e}}_{\mathrm{p}} n[Q \mid R']$. By Lemma 5.5 and Theorem 3.17 it follows $n[P \mid R'] \approx n[Q \mid R']$. As a consequence, $n[P \mid R'] \;\mathcal{R}\; n[Q \mid R']$.

□

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399